



[www.
ideca.
gov.
co](http://www.ideca.gov.co)

Documento técnico descriptivo de los formatos de intercambio J-SON, GEOJ-SON y el protocolo SOAP

Bogotá, D.C., 24/12/2020



Documento técnico descriptivo de los formatos de intercambio J-SON, GEOJ-SON y el protocolo SOAP

Gerente IDECA

Julia Espindola García

Subgerente de Operaciones

Pedro Alberto Pinzón Montero

Profesional Especializado Gerencia IDECA

Astrid Yadira Duitama Guio

Equipo de Trabajo

Profesional Especializado Gerencia IDECA

Sandra Duran Duran



Fecha de creación o actualización: 18/12/2020

Página web: www.ideca.gov.co

Correo electrónico: ideca@catastrobogota.gov.co

Licencia: Attribution 4.0 International (CC BY 4.0)



ALCALDÍA MAYOR
DE BOGOTÁ D.C.

UAECD
Catastro Bogotá



Control de Versiones

CAMBIOS

Fecha	Autor	Versión	Cambio efectuado
18/12/2020	Astrid Yadira Duitama	1.0	Primera versión del documento. No hay cambios para registrar.

REVISORES

Nombre	Dependencia
Sandra Duran Duran	Gerencia IDECA

Tabla de Contenido

Introducción.....	6
Objetivo.....	7
Alcance.....	8
Términos, Siglas y Abreviaturas.....	9
Descripción JSON.....	10
Gramática.....	11
Valores.....	12
Objetos.....	12
Arreglos.....	12
Números.....	13
Cadenas de caracteres (string).....	13
Descripción GeoJSON.....	14
Objeto GeoJSON.....	15
Geometría.....	15
Feature (Objeto Geográfico).....	19
FeatureCollection (Contenedor de Feature).....	20
Sistemas de Referencia.....	20
Descripción SOAP.....	21
Conceptos Previos.....	21
Sistemas Distribuidos.....	21
Arquitectura Orientada a Servicios (SOA).....	21
SOA con servicios web.....	23
Interoperabilidad.....	24
Mensajería de servicios web.....	24
Terminología SOAP.....	25
Protocolo de Arquitectura Orientada a Servicios -SOAP.....	26
Referencias.....	28

Tabla de Ilustraciones

Ilustración 1 Estructura del Objeto.....	15
Ilustración 2, Estructura Objeto Punto.....	16
Ilustración 3, Estructura Objeto Multipunto	16
Ilustración 4, Estructura Objeto Línea.....	17
Ilustración 5, Estructura Objeto Multilínea.....	17
Ilustración 6, Estructura Objeto Polígono.....	18
Ilustración 7, Estructura Objeto Multipolígono	18
Ilustración 8, Estructura Objeto Colección.....	19
Ilustración 9, Elemento Geometría	19
Ilustración 10, Elemento Característica.....	20
Ilustración 11, Estructura SOAP	26

Introducción

Partiendo de la definición de una Infraestructura de Datos Espaciales -IDE- como un conjunto de datos y recursos geográficos, estándares, políticas, tecnologías y acuerdos institucionales que, de forma integrada y sostenida, facilitan la producción, disponibilidad y acceso a la información geográfica de manera articulada entre varias entidades productoras, donde uno de los principales retos es la distribución, conservación y disponibilidad de datos a todos los usuarios y en todo lugar.

Una solución para el trabajo o edición de los datos es permitir la descarga, sin embargo, esto no garantiza la visualización actualizada de los mismos, como solución a esto nacen los formatos ligeros con capacidades que permiten su uso en línea, los cuales garantizan la interoperabilidad en diferentes plataformas y programas, además de una disposición descentralizada de los datos.

Dentro de los formatos más utilizados en IDECA están JSON y GeoJSON; JSON de las siglas (JavaScript Object Notation) es un formato de intercambio de datos ligero, basado en texto, con sintaxis de lenguaje independiente, fácil de leer y escribir para los humanos, mientras que para las máquinas de generar e interpretar, basado en un subconjunto del estándar de lenguaje de programación JavaScript. JSON es un formato de texto que es completamente independiente a los lenguajes de programación, el cual define un pequeño conjunto de reglas de estructuración para la representación portátil de datos estructurados, pero a su vez utiliza convenciones que son ampliamente conocidas por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un formato ideal para el intercambio de datos.

Por su lado GeoJSON es un formato de intercambio de datos geoespaciales basado en JSON, el cual define varios tipos de objetos y la representación conjunta de datos con características geográficas, sus atributos y extensión espacial.

Otra forma de intercambio de datos es por medio de servicios web, esta tecnología busca ofrecer servicios disponibles en cualquier lugar, momento y plataforma, basado en un conjunto de protocolos y estándares que garanticen este intercambio entre diferentes aplicaciones, uno de los protocolos de intercambio existentes en IDECA es SOAP, inicialmente de las siglas (Simple

Object Access Protocol) y ahora (Service Oriented Architecture Protocol) dependiendo de la interpretación de la tecnología¹, reglamentado por el comité de normalización W3C, el cual define un marco estándar, extensible y compatible para la comunicación entre aplicaciones por medio del intercambio de datos en lenguaje XML.

Objetivo

Facilitar la implementación y uso de los formatos JSON, GeoJSON y del protocolo SOAP mediante un documento de descripción técnica para todos los usuarios y productores de información geográfica de los miembros o usuarios de IDECA.

¹ https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#service_oriented_architecture

Alcance

El presente documento no es un estándar o especificación, es una descripción técnica, para que la comunidad IDECA apropie los formatos de intercambio de datos y protocolo más utilizados.

Está basado en las especificaciones técnicas ECMA 404 del 2017 y IETF 8259 de 2017 para JSON; el RFC 7946 de IETF para GeoJSON, entidades responsables del mantenimiento y generación de dichos estándares, por lo que este documento no es superior a ellos y puede quedar obsoleto al presentarse cambios. Es responsabilidad del usuario estar atento a posibles actualizaciones.

ECMA International es una asociación industrial fundada en 1961, dedicada a la estandarización de sistemas de información y comunicación.² El Grupo de Trabajo de Ingeniería de Internet (IETF) es el principal organismo de estándares de Internet, que desarrolla estándares abiertos a través de procesos abiertos³.

Para el caso del protocolo SOAP la entidad reguladora de los servicios Web es la W3C, comunidad internacional que desarrolla estándares abiertos para asegurar el crecimiento de la web a largo plazo, y el documento base es “SOAP Versión 1.2 Parte 0: Introducción (Segunda edición)” el cual es un documento destinado a proporcionar un tutorial fácilmente comprensible sobre las características de SOAP, específicamente describe las características a través de varios escenarios de uso.⁴

² <https://www.ecma-international.org/>

³ <https://www.ietf.org/about/>

⁴ <https://www.w3.org/TR/2007/REC-soap12-part0-20070427/>

Términos, Siglas y Abreviaturas

Unicode: Es el estándar de codificación de caracteres universal utilizado para la representación de texto para procesamiento del equipo. Unicode proporciona una manera consistente de codificación de texto multilingüe y facilita el intercambio de archivos de texto internacionales.⁵

Tabla hash: Es una estructura de datos que utiliza una función hash para identificar datos mediante una llave o clave (ej. Nombre de una persona). La función hash transforma una llave a un valor índice de un arreglo de elementos.

OGC: El Consorcio Geoespacial Abierto (OGC) es una asociación internacional de empresas, agencias gubernamentales, organizaciones de investigación y universidades impulsadas para hacer que la información y los servicios geoespaciales (ubicación) sean FAIR: localizables, accesibles, interoperables y reutilizables.

Feature: Objeto geográfico, abstracción de un fenómeno del mundo real⁶

Protocolo: En informática y telecomunicaciones, hablamos de un protocolo para referirnos a un sistema de normas que regulan la comunicación entre dos o más sistemas que se transmiten información a través de diversos medios físicos. Dicho en otras palabras, los protocolos son lenguajes o códigos de comunicación entre sistemas informáticos, definidos en base a una sintaxis, una semántica y una sincronización, así como de métodos de recuperación de errores. De los protocolos depende, entonces, hacer que dos o más sistemas informáticos puedan comunicarse de manera eficaz y ordenada, es decir, que hablen un mismo idioma. Estos lenguajes se implementan mediante hardware o software, o combinaciones de ambos, y le brinda a cada participante en la comunicación una identidad y un método específico de procesamiento de la información⁷.

⁵ https://docs.oracle.com/cd/E26921_01/html/E27143/glmgn.html

⁶ The ISO/TC 211 Multi-Lingual Glossary of Terms, version junio del 2020

⁷ <https://concepto.de/protocolo-informatico/#ixzz6fxJPtd9s>

Descripción JSON

JSON es un formato de texto que facilita el intercambio de datos estructurados entre todos los lenguajes de programación. La sintaxis consta de llaves, corchetes, dos puntos y coma, la cuál es útil en muchos contextos, perfiles y aplicaciones. JSON corresponde a las siglas de JavaScript Object Notation inspirado en los objetos literales de JavaScript, también conocido como ECMAScript, como se define en la Especificación del lenguaje ECMAScript, tercera edición⁸.

Sin embargo, no intenta imponer las representaciones de datos internos de ECMAScript en otros lenguajes de programación. En cambio, comparte un pequeño subconjunto de la sintaxis de ECMAScript con todos los demás lenguajes de programación. La sintaxis de JSON no es una descripción para el intercambio completo de datos, el intercambio significativo de estos requiere un acuerdo entre un productor y un consumidor. Lo que proporciona JSON es el marco sintáctico al que se puede adjuntar dicha semántica.

En cualquier lenguaje de programación, puede haber una variedad de números de diversas capacidades y complementos, fijos o flotantes, binarios o decimales. Que puede hacer difícil el intercambio entre diferentes lenguajes de programación, JSON en cambio ofrece solo la representación de números usada por los humanos: es decir una secuencia de dígitos. Todos los lenguajes de programación saben cómo entender las secuencias de dígitos incluso si no están de acuerdo con las representaciones internas. Eso es suficiente para permitir el intercambio.

Los lenguajes de programación varían ampliamente dependiendo del soporte que requieren los objetos, es decir de las características y limitaciones que estos presentan, en consecuencia, los modelos de sistemas de objetos pueden ser muy divergentes y estar en continua evolución. JSON, en cambio, proporciona una notación simple para expresar las colecciones de los pares: nombre / valor. La mayoría de los lenguajes de programación tendrán alguna característica para representar dichas colecciones, denominadas como: registro, estructura, diccionario, mapa, tabla hash u objeto.

JSON también proporciona soporte para listas ordenadas de valores. Todos los lenguajes de programación tendrán alguna característica para representar dichas listas, denominadas como: arreglo, vector o lista. Dado que los objetos y arreglos se pueden anidar, árboles y otras estructuras complejas de datos pueden ser

⁸ ECMA-262, ECMAScript® Language Specification

representadas. Al ser aceptada la convención simple de JSON, las estructuras de datos complejas pueden ser fácilmente intercambiables entre lenguajes de programación incompatibles.

JSON no admite gráficos cíclicos, al menos no directamente. JSON no está indicado para aplicaciones que requieren datos binarios.

Debido a su simpleza no se espera que la gramática JSON llegue a cambiar. Esto le da a JSON una notación fundamental altamente estable.

Gramática

Un texto JSON es un valor serializado, es decir una estructura de datos en secuencia que puede ser enviada a través de un flujo⁹, enmarcadas con claves de seguridad (tokens), el conjunto de tokens incluye seis estructurales, cadenas, números y tres de nombre literal. Vale la pena aclarar que algunas especificaciones consideran el texto JSON como un objeto o un arreglo.

- Los 6 tokens estructurales son:

Símbolo	Función	Nombre
[Comenzar Arreglo	Corchete izquierdo
{	Comenzar Objeto	Llave izquierda
]	Fin Arreglo	Corchete derecho
}	Fin Objeto	Llave derecha
:	Separador de nombre	Dos puntos
,	Separador de valor	Coma

- Los 3 tokens de nombre literal son: **true**, **false** y **null**

Serán permitidos los espacios en blanco antes o después de cualquier token. Se considera espacio en blanco cualquier secuencia de uno o más de los siguientes:

⁹ <http://www.jtech.ua.es/j2ee/publico/lja-2012-13/sesion05-apuntes.html>

Tabulación, salto de línea, retornar al inicio de línea y espacio. No son permitidos los espacios en blanco dentro del token.

Valores

Un valor JSON es: un objeto, matriz, número, cadena de caracteres, o uno de los siguientes tres nombres literales los cuales deben estar en minúscula: true, false o null.

Objetos

La estructura del objeto es representada con un par de llaves que encierran a cero o más parejas: nombre/valor, siendo nombre una cadena de valores, seguido del token dos puntos (:) que separa el nombre del valor. Una coma los separa de la siguiente pareja.

Se recomienda que los nombres dentro de un objeto no se repitan para garantizar la interoperabilidad en el sentido que todas las implementaciones de software que reciben ese objeto entiendan las asignaciones de nombre/valor. De no ser así el comportamiento del software receptor es impredecible, muchas implementaciones informan solo el último par nombre/valor, otras informan un error o no pueden analizar y otras reportan todos los pares de nombre/valor, incluidos los duplicados.

Se ha observado que las bibliotecas de análisis JSON difieren en cuanto a si hacen visible o no el orden de los miembros del objeto para el software de llamada. Las implementaciones cuyo comportamiento no dependa del orden de los miembros serán interoperables en el sentido de que no se verán afectadas por estas diferencias.

Arreglos

La estructura del arreglo consta de un par de corchetes que encierran cero o más valores. Los valores están separados por comas. La sintaxis JSON no define un sentido específico según el orden de los valores, sin embargo, la estructura del arreglo se usa a menudo en situaciones en las que hay algo de orden semántico. No es necesario que los valores en un arreglo sean del mismo tipo.

Números

La representación de los números es similar a la que se utiliza en la mayoría de los lenguajes de programación, un número es una secuencia decimal de dígitos sin el cero adicional a la izquierda. Puede tener un signo menos antes, una parte fraccionaria después del punto decimal, un exponente prefijado por e o E con su respectivo signo (+/-).

No son permitidos los valores numéricos que no se puedan representar como secuencia de dígitos.

Cadenas de caracteres (string)

Una cadena es una secuencia de caracteres entre comillas, la cual es similar a la utilizada en la familia de lenguajes de programación C. Todos los caracteres pueden colocarse entre comillas, excepto: la barra inversa, las comillas y los caracteres de control (U + 0000 a U + 001F), correspondiente a las dos primeras filas de la tabla Unicode¹⁰.

De ser necesaria la representación de un carácter de la excepción, este deberá ser representado por los 6 dígitos hexadecimales, dentro de las comillas, así por ejemplo el carácter barra inversa (\), será representado como "U005C".

¹⁰ <https://unicode-table.com/es/#basic-latin>

Descripción GeoJSON

GeoJSON es un formato de intercambio de datos geoespaciales basado en JSON el cual define varios tipos de objetos y la forma combinada para representar datos con características geográficas, sus atributos y extensión espacial.

Un objeto GeoJSON puede representar una región del espacio (Geometría), una entidad limitada espacialmente (Feature), o una lista de Features (FeatureCollection). Los tipos de geometría que soporta son: Punto, Línea, Polígono, Multipunto, Multilínea, Multipolígono y Colección Geométrica. un Feature se define como un Objeto geográfico con propiedades adicionales, y FeatureCollection es una lista de features.

El formato se ocupa en un sentido más amplio de los datos geográficos, de esta manera, cualquier elemento delimitado en el espacio que tenga atributos podría ser un Feature, sea o no una estructura física. Estos conceptos no son nuevos; se derivan de estándares abiertos preexistentes en los sistemas de información geográfica y se han simplificado para ser adaptados al desarrollo de aplicaciones web usando JSON.

GeoJSON comprende los siete tipos de geometría definidos por la OGC, y sus respectivas dimensiones que pueden ser: 0, 1 o 2.

- Dimensión 0: para una geometría sin longitud ni área es decir *Punto* y *Multipunto*.
- Dimensión 1: para una geometría sin área, *Línea* y *Multilínea*.
- Dimensión 2: para una geometría con área diferente de cero, *Polígono* y *Multipolígono*.
- y el heterogéneo *Colección geométrica* que es una geometría que consiste en una colección de una o más geometrías de cualquier clase; quien tendrá las mismas dimensiones de los elementos que los componen.

La estructura de un objeto GeoJSON se muestra a continuación, más adelante se detallará lo referente a objeto, y coordenadas.

11

Ilustración 1 Estructura del Objeto
Fuente: Elaboración propia

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```

Objeto GeoJSON

Un objeto GeoJSON puede ser de tipo: Geometría, un feature, o una colección de Features, el cual debe ser citado al comienzo de la estructura.

Geometría

Cada objeto geométrico es un objeto GeoJSON, sin importar donde ocurra dentro del texto y tendrá que definirse el tipo (Punto, Línea, Polígono, Multipunto, Multilínea, Multipolígono y Colección Geométrica), siempre debe ir acompañado de un arreglo con las coordenadas.

La ubicación es fundamental en la construcción de la geometría, es un arreglo de números que debe tener dos o tres elementos separados por comas, con números decimales correspondientes a longitud y latitud, o este y norte en ese orden estricto, la altitud o elevación puede incluirse en la tercera posición del elemento siempre que no exceda las tres posiciones.

Una línea entre dos ubicaciones siempre corresponderá a la distancia más corta entre ellas, una línea recta en caso de las coordenadas planas y una línea geodésica en el caso de las coordenadas geográficas. Tenga en cuenta que esta línea puede diferir notablemente de la ruta geodésica a lo largo de la superficie curva del elipsoide de referencia.

Se utilizará Multipunto, Multilínea o Multipolígono cuando el objeto geográfico tenga más de una parte física y solo haga referencia a un atributo, como las sedes de una entidad Bancaria o la bifurcación de un drenaje.

Teniendo en cuenta el tipo de geometría el objeto tendrá una representación coordenada así:

- **Punto**

Para el tipo punto el elemento será una única ubicación.

12

Ilustración 2, Estructura Objeto Punto
Fuente: Elaboración propia

```
{  
  "type": "Point",  
  "coordinates": [ 32.34375, 48.4583518828086 ]  
}
```

- **Multipunto**

Para el tipo Multipunto el elemento será un arreglo de ubicaciones.

13

Ilustración 3, Estructura Objeto Multipunto
Fuente: Elaboración propia

```
{  
  "type": "MultiPoint",  
  "coordinates": [ 32.34375, 48.4583518 ], [ 32.4565, 48.98767 ],  
  [ 32.76576, 49.097898 ]  
}
```


- Línea

Para el tipo línea el elemento será un arreglo de dos o más ubicaciones.

14

Ilustración 4, Estructura Objeto Línea
Fuente: Elaboración propia

```
{
  "type": " LineString",
  "coordinates": [
    [ 32.34375, 48.4583518 ], [ 32.4565, 48.98767 ], [ 32.76576,
    49.097898]
  ]
}
```

- Multilínea

Para el tipo Multilínea el elemento será un arreglo de arreglos de coordenadas de línea.

15

Ilustración 5, Estructura Objeto Multilínea
Fuente: Elaboración propia

```
{
  "type": "MultiLineString",
  "coordinates": [
    [
      [ 32.34375, 48.4583518 ], [ 32.4565, 48.98767 ]
    ], [
      [ -32.76576,- 49.097898] , [ -32.4565,- 48.98767 ]
    ]
  ]
}
```

- Polígono

Para hablar del polígono es útil introducir el concepto de anillo lineal.

El *anillo lineal* es un polígono cerrado con cuatro o más ubicaciones, donde la primera y última debe tener valores idénticos, este anillo corresponderá al límite de una superficie o un agujero de superficie.

Aunque el anillo no es un tipo de geometría representada en GeoJSON, conduce a la definición geométrica del tipo polígono.

El elemento coordenadas será una matriz de matrices de anillos lineales, para polígonos con más de un anillo, el primero corresponde al anillo exterior y los demás serán considerados agujeros interiores.

16

 Ilustración 6, Estructura Objeto Polígono
 Fuente: Elaboración propia

```
{
  "type": "Polygon",
  "coordinates": [
    [ 32.34375, 48.4583518], [ 32.4565, 48.98767 ], [ 32.76576,
    49.097898] , [ 32.34375, 48.4583518]
  ]
}
```

- **MultiPolígono**

Para multipolígono el elemento de coordenadas será una matriz de matrices de polígonos.

17

 Ilustración 7, Estructura Objeto Multipolígono
 Fuente: Elaboración propia

```
{
  "type": "MultiPolygon",
  "coordinates": [
    [
      [ 32.34375, 48.4583518], [ 32.4565, 48.98767], [ 32.34375,
      48.4583518]
    ], [
      [ -32.76576, - 49.097898], [ -32.4565, 48.98767 ], [ -32.76576, -
      49.097898]
    ]
  ]
}
```

- **Colección Geométrica**

A diferencia de los objetos geométricos descritos anteriormente, la colección geométrica puede estar compuesta por diferentes tipos, por ejemplo, punto y línea.

Para esta geometría la sintaxis varia, en cambio del elemento “coordenadas” se define un elemento tipo arreglo llamado “geometrías”, donde cada componente será un objeto geométrico (punto o línea, etc.) que contendrá sus coordenadas, describiendo así las partes de esta composición.

18

 Ilustración 8, Estructura Objeto Colección
 Fuente: Elaboración propia

```

{
  "TYPE": "GeometryCollection",
  "geometrías": [
    {
      "type": "Point",
      "coordinates": [ 32.34375, 48.4583518828086]},
    {
      "type": " LineString",
      "coordinates": [
        [ 32.34375, 48.4583518], [ 32.4565, 48.98767 ], [ 32.76576,
        49.097898]
      ]
    }
  ]
}
    
```

Feature (Objeto Geográfico)

Un Feature representa una cosa limitada espacialmente, será un objeto GeoJSON sin importar donde ocurra dentro del texto, debe definirse como Tipo “Feature”, y estará incluido el elemento “geometría”, descrito anteriormente, con sus respectivas coordenadas. También contará con el elemento “propiedades” como otro objeto donde se incluirán los atributos de tenerlos.

19

 Ilustración 9, Elemento Geometría
 Fuente: Elaboración propia

```

{
  "type": "Feature",
  "properties": {"id": 1, "NombreAtributo": "azul" },
  "geometry": {
    "type": "Polygon",
    "coordinates": [ [ x1, y1], [x2, y2], [x3,y3], [x1,y1] ]
  ]
}
    
```

FeatureCollection (Contenedor de Feature)

Un FeatureCollection es un objeto GeoJSON, que contendrá los objetos geográficos del archivo GeoJSON, adicional a los elementos descritos anteriormente se incluirá el elemento “característica” de tipo arreglo.

I 10

 Ilustración 10, Elemento Característica
 Fuente: Elaboración propia

```
{
  "type": "FeatureCollection",
  "name": "Localidades",
  "crs": { "type": "name", "properties": { "name":
    "urn:ogc:def:crs:EPSG::4686" }
  },
  "type": "Feature",
  "properties": { "id": 1, "NombreAtributo": "azul" },
  "geometry": {
    "type": "Polygon",
    "coordinates": [ [ x1, y1], [x2, y2], [x3,y3], [x1,y1] ]
  }
}
```

Es usado generalmente para la creación de capas o datos geográficos, en el ejemplo es la capa de Localidades.

Sistemas de Referencia

El sistema de referencia de coordenadas para los datos GeoJSON es por defecto el sistema geodésico mundial Dátum del Sistema 1984 (WGS 84), con unidades de longitud y latitud de grados decimales. Equivalente a la referencia de coordenadas sistema identificado por el Consorcio Geoespacial Abierto (OGC) URN cómo - urn: ogc: def: crs: OGC :: CRS84-.

Para el uso de sistemas de referencia alternativos no se puede garantizar la interoperabilidad, a menos que las partes coordinadas tengan un arreglo previo para leer los datos sin el riesgo de ser mal interpretados.

Esta descripción será incluida en el elemento de características descrito anteriormente.

Descripción SOAP

Conceptos Previos

Algunos de los conceptos necesarios para comprender el protocolo SOAP son los sistemas distribuidos, la arquitectura orientada a servicios, el servicio Web, interoperabilidad y mensajería.

Sistemas Distribuidos

Un Sistema Distribuido está compuesto de varios agentes de software unitarios que deben trabajar juntos para realizar algunas tareas, estos agentes del sistema distribuido no operan en el mismo entorno de procesamiento, por lo que deben comunicarse mediante protocolos de hardware/software a través de una red, esto significa que las comunicaciones con un sistema distribuido son básicamente menos rápidas e íntegras que las que utilizan la solicitud directa de código y la memoria compartida. Esto implica que la arquitectura de los sistemas distribuidos requiere que los desarrolladores (de infraestructura y aplicaciones) consideren la latencia impredecible del acceso remoto y tengan en cuenta los problemas de concurrencia y la posibilidad de fallas parciales.

Por otra parte, los sistemas de Objetos Distribuidos exponen mediante reglas semánticas sistemas remotos para transar solicitudes, generalmente se caracterizan por la interacción del objeto y el programa que lo usa.

Arquitectura Orientada a Servicios (SOA)

Una arquitectura orientada a servicios (SOA) describe un entorno donde las aplicaciones de software exponen la funcionalidad a través de una interfaz de mensajería del proveedor de servicios. Otros agentes de software actúan como consumidores de servicios al utilizar la funcionalidad expuesta por los proveedores. Las aplicaciones de software pueden actuar tanto como consumidores de servicios como proveedores de estos al proporcionar funcionalidad al usuario. Por ejemplo, la aplicación de teléfono de un agente de viajes puede ayudar al usuario a planificar un viaje accediendo a un servicio de aerolínea (para reservar un vuelo) y a un servicio de reserva de hotel (para reservar una habitación). Por lo tanto, el software de viaje del teléfono consumió dos servicios para proporcionar un nuevo servicio (reservar un viaje).

Una arquitectura orientada a servicios (SOA) es una forma de arquitectura de sistemas distribuidos que normalmente se caracteriza por las siguientes propiedades:

- **Vista lógica:** El servicio es una vista lógica y abstraída de programas, bases de datos, procesos de negocio, etc., definidos en términos de lo que hace, normalmente llevando a cabo una operación de nivel empresarial.
- **Orientación del mensaje:** El servicio se define formalmente en términos de los mensajes intercambiados entre los agentes proveedores y agentes solicitantes no de las propiedades de los agentes como tal. La estructura interna de un agente, incluidas características como el lenguaje de implementación, la estructura de procesos e incluso la estructura de la base de datos, se abstraen deliberadamente en la SOA: al utilizar la arquitectura SOA, no es necesario saber cómo se está implementando un servicio de un agente. Uno de los beneficios clave son los llamados sistemas heredados, que evitan cualquier conocimiento de la estructura interna de un agente, es posible incorporar cualquier componente de software o aplicación que se pueda "envolver" en un código de manejo de mensajes que le permita adherirse a la definición formal del servicio.
- **Orientación descriptiva:** Un servicio se describe mediante metadatos procesables por máquina, esta descripción respalda la naturaleza pública de la SOA, solo deben incluirse en la descripción aquellos detalles que están expuestos al público e importantes para el uso del servicio. La semántica de un servicio debe documentarse, directa o indirectamente, mediante su descripción.
- **Granularidad:** Los servicios tienden a utilizar una pequeña cantidad de operaciones con mensajes relativamente grandes y complejos.
- **Orientación a la red:** Los servicios tienden a orientarse hacia el uso a través de una red, aunque esto no es un requisito absoluto.
- **Plataforma neutral:** Los mensajes se envían en un formato estandarizado y neutral para la plataforma entregado a través de las interfaces. XML es el formato más conocido que cumple con esta restricción.

Un SOA, está conformado principalmente por 3 componentes:

- **Descubrimiento**

Antes de que una parte (solicitante) pueda contactar a otra (proveedor) para obtener servicios, el solicitante debe localizar un proveedor apropiado. Este proceso se conoce como descubrimiento. Se pueden utilizar diferentes

mecanismos en el proceso, como buscar servicios enumerados en un sitio web, buscar en un directorio o ubicar al proveedor a través de un marco que admita el proceso.

- **Descripción**

Una vez que se ha localizado al proveedor, el solicitante debe aprender reglas relacionadas con él (por ejemplo, qué protocolos usar para estructurar el mensaje, qué seguridad y otras políticas debe prestar atención). Este proceso se conoce como descripción. La información de descripción puede ser devuelta como parte del proceso de descubrimiento. La descripción típicamente incluye información estructurada, así como otra información.

- **Mensajería**

El tercer paso es comunicarse con el proveedor enviando (y posiblemente recibiendo) mensajes. La comunicación suele tener lugar utilizando el protocolo SOAP junto con protocolos inferiores como HTTP. La definición de cómo se utilizará un protocolo inferior se denomina enlace SOAP. Es necesario para lograr la interoperabilidad entre los mensajes SOAP y su protocolo inferior.

SOA con servicios web

Un servicio web es un sistema de software diseñado para soportar la interacción interoperable de máquina a máquina a través de una red. Cuenta con una interfaz descrita en un formato procesable por máquina (específicamente WSDL). Otros sistemas interactúan con el servicio web de la manera establecida por los mensajes SOAP, que habitualmente se transmiten mediante HTTP con una serialización XML junto con otros estándares relacionados con la web¹¹.

Este conjunto específico de tecnologías se puede utilizar para implementar una SOA, aunque estas también pueden crearse utilizando otras tecnologías, los servicios web se están adoptando ampliamente. Una de las razones del éxito de los servicios web es el beneficio inmediato que aportan a los proyectos de integración de sistemas. La tecnología de servicios web se puede utilizar para envolver sistemas heredados y reducir el costo de integración, lo que produce ventajas financieras inmediatas. Estas ventajas incluyen la eliminación de datos

¹¹ Definición servicio WEB w3c: <https://www.w3.org/TR/ws-gloss/>

duplicados, evitar los errores y costos involucrados en volver a ingresar datos, así como las ventajas competitivas de integrar sistemas, recursos e información.

Los principales estándares relacionados con los servicios web se centran en la mensajería, la descripción y los directorios que permiten el desarrollo de implementaciones que son independientes de la arquitectura, de la plataforma informática, el sistema operativo, el lenguaje de la computadora y la interfaz de programación. Sin embargo, esto no impide que la estandarización relacionada de las interfaces de programación para plataformas específicas admita servicios web, los estándares básicos detrás de los servicios web, como XML, HTTP, SOAP, WSDL y XML Security, han sido normalizados por el W3C.

Interoperabilidad

La razón principal del éxito de los servicios web es que el método para habilitar la interoperabilidad no es una interfaz de programación estándar, sino un formato estándar para un protocolo de mensajería.

Esto permite la creación de sistemas poco acoplados independientemente del lenguaje de programación, plataforma de hardware, sistema operativo o entorno de desarrollo. El único requisito es que los sistemas puedan producir y consumir mensajes de protocolo estándar. Un sistema tan poco acoplado es modular, lo que facilita la modificación de partes individuales de forma independiente (por ejemplo, para agregar nuevos servicios a un flujo de mensajes, reemplazar implementaciones de servicios o incluso reconfigurar aplicaciones completas). Como resultado, es posible lograr la interoperabilidad en una amplia gama de sistemas, con un impacto mínimo en las implementaciones. Un ejemplo es que uno o más sistemas minoristas acceden a uno o más almacenes que reponen sus existencias de uno o más fabricantes, se demuestra la interoperabilidad de la aplicación al hacer que diferentes proveedores proporcionen diferentes implementaciones para los sistemas de minoristas, almacenes y fabricantes, todas ejecutándose como servicios en Internet.

Mensajería de servicios web

Dado que las aplicaciones orientadas a servicios se crean a partir de componentes que se envían mensajes entre sí, la mensajería constituye un aspecto central de los servicios web. Otros aspectos pueden manejarse fuera de banda, mientras que el componente de mensajería es fundamental. La adopción de servicios web comenzó a partir de aplicaciones simples que requerían mensajería para lograr la interoperabilidad. Al comenzar, los desarrolladores de servicios web pueden

posponer los problemas relacionados con la descripción, el descubrimiento, la seguridad y la confiabilidad, y abordarlos cuando sea necesario. La mensajería requiere un método para dirigir los mensajes a los puntos finales correctos. Una URL es parte de la información de dirección, pero a menudo el receptor requiere información adicional para enviar y procesar el mensaje correctamente.

Terminología SOAP

- **Nodo SOAP:** Se refiere a la ejecución del procesamiento lógico necesario para transmitir, recibir, procesar o retransmitir un mensaje. Un nodo SOAP es el responsable de hacer cumplir las reglas que rigen el intercambio y acceder a los servicios proporcionados por los protocolos subyacentes a través de uno o más enlaces SOAP.
- **Rol SOAP:** La función esperada de un receptor SOAP al procesar un mensaje. Un receptor SOAP puede actuar en múltiples roles.
- **Enlace SOAP:** El conjunto formal de reglas para llevar un mensaje SOAP dentro de otro protocolo (protocolo subyacente) con el propósito de intercambio.
- **Mensaje SOAP:** La unidad básica de comunicación entre nodos SOAP.
- **Envoltura SOAP:** Es el elemento de información más externo de un mensaje SOAP.
- **Encabezado SOAP:** Una colección de cero o más bloques de encabezado SOAP, cada uno de los cuales podría estar dirigido a cualquier receptor SOAP dentro de la ruta del mensaje.
- **Bloque de encabezado SOAP:** Un elemento de información de elementos que se utiliza para delimitar datos que lógicamente constituyen una única unidad computacional dentro del encabezado. El tipo bloque de encabezado SOAP se identifica mediante el nombre expandido XML del elemento de información del elemento del bloque del encabezado.
- **Cuerpo SOAP:** Una colección de cero o más elementos de información de elementos destinados a un receptor final en la ruta del mensaje.

Protocolo de Arquitectura Orientada a Servicios - SOAP

SOAP es fundamentalmente una estructura definida para el intercambio de información, donde la carga útil ocupará el cuerpo del mensaje y la información adicional o de control estará en el encabezado.

Un mensaje SOAP es una transmisión unidireccional entre nodos, desde un remitente a un receptor, sin embargo, se espera que las aplicaciones implementen patrones de interacción más complejos que van desde: solicitud/respuesta, solicitud/respuestas múltiples, hasta intercambios de ida y vuelta “conversaciones”.

I 11

Ilustración 11, Estructura SOAP
Fuente: Basada en libro: Servicios web móviles



El elemento encabezado es opcional y es un mecanismo de extensión que proporciona una forma de enviar información adicional diferente a la carga útil del mensaje. Tal información de “control” puede incluir, directivas o información contextual relacionada con el procesamiento, la seguridad, o confiabilidad del mensaje. Esto permite extender un mensaje SOAP de una manera específica a la aplicación.

Los encabezados pueden ser inspeccionados, insertados, eliminados o reenviados por los nodos encontrados a lo largo de una ruta de mensaje, es importante

aclarar que la especificación W3C no se refiere al contenido de los elementos dentro del encabezado, ni a cómo se enrutan los mensajes SOAP entre nodos, ni a la forma en que se determina la ruta, etc.

Los encabezados han sido diseñados anticipándose a varios usos de SOAP, muchos de los cuales implicarán la participación de otros nodos de procesamiento llamados intermediarios, a lo largo de la ruta de un mensaje desde un remitente hasta un receptor, lo cual permite a los intermediarios proporcionar servicios de valor agregado.

Los elementos secundarios inmediatos, bloques de encabezado, representan una agrupación lógica de datos que, pueden dirigirse individualmente a los nodos SOAP y podrían encontrarse en la ruta de un mensaje desde un remitente hasta un receptor final. Tenga en cuenta que siempre que se procesa un bloque de encabezado, debe hacerse de acuerdo con las especificaciones para ese bloque. Estas especificaciones de bloques de encabezado están definidas por la aplicación y no forman parte de SOAP.

El cuerpo SOAP es el elemento obligatorio dentro de la envoltura, lo que implica que es aquí donde debe transportarse la información principal de extremo a extremo, transmitida en un mensaje SOAP, generalmente esta información está estructurada en XML debido a las ventajas que ofrece el lenguaje, sin embargo, podría incluirse un mensaje SOAP anidado, para el envío de archivos adjuntos, los cuales no requieran un procesamiento XML, como una imagen o un documento, el resultado sería una reducción de tiempo en procesamiento en el análisis inmediato.

Elegir qué datos incluir en el bloque de encabezado y cuales en el cuerpo de SOAP son decisiones que se toman en el momento del diseño de la aplicación, lo principal es tener en cuenta que los bloques de encabezado pueden dirigirse a varios nodos ubicados a lo largo de la ruta de un mensaje, desde un remitente hasta el destinatario final, dichos nodos SOAP intermedios pueden proporcionar servicios de valor añadido basados en los datos de dichos encabezados.

Referencias

Introducción al JSON, disponible en: <http://www.json.org/json-es.html>

The JavaScript Object Notation (JSON) Data Interchange Format, disponible en: <https://tools.ietf.org/html/rfc8259>

The GeoJSON Format, disponible en: <https://tools.ietf.org/html/rfc7946>

The JSON Data Interchange Syntax, disponible en: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>,

ECMA organización, disponible en: <https://www.ecma-international.org/ecma-262/11.0/index.html#title>

Descripción general de Unicode,
https://docs.oracle.com/cd/E26921_01/html/E27143/glmgn.html y lista de
códigos <https://unicode-table.com/es/>

Open GIS Consortium, Inc. OpenGIS; Simple Features Specification For SQL
Revision 1.1

SOAP Protocol; <https://www.w3.org/TR/soap/>

Glosario W3C, <https://www.w3.org/TR/ws-gloss/>

https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#service_oriented_architecture

Servicios web móviles: arquitectura e implementación, Frederick Hirsch, John
Kemp, Jani Ilkka, Marzo de 2006

