

Catálogo de recursos para los datos abiertos

Domingo Ortega Pérez de Villar

Resumen

Se ha realizado un estudio y revisión bibliográfica relativa a los recursos abiertos que ofrecen gobiernos, instituciones, y empresas del sector privado. El objetivo ha sido el de ver cómo se ofrecen, qué herramientas hay para acceder a ellos, y cómo recogerlos para poder trabajar con ellos. Se analiza en el entorno de las WebAPIs para ver cómo se pueden desarrollar y documentar éstas para trabajar con estos datos.

El presente estado del arte tiene como objetivos: (1) presentar dónde nos encontramos en lo referente a la publicación de esta información, (2) mostrar la problemática y posibles preocupaciones que puedan haber, (3) exponer cómo se presentan estos datos, (4) exponer el modelo REST, (5) los lenguajes que se utilizan para interactuar con la información y nombrar herramientas para trabajar con ellos, y (6) cómo documentar las WebAPIs.

Introducción

Actualmente se hace cada vez más difícil pensar en un negocio, u organización, que exponga información en la Web y que no tenga un fin definido para ésta. Cada vez más se está reconociendo y dando importancia a la publicación de datos, para mostrar transparencia y para ayudar a negocios afines a dicha información. Por eso también, cada vez más, ofrecen una serie de servicios para que otros, o ellos mismos, puedan trabajar con dicha información. Estos servicios son accesibles a través de APIs que publican, y pretenden que sean lo más sencillas y útiles posible. Pretenden así ofrecer una imagen de cercanía y usabilidad de los distintos servicios. Una API es tan buena como lo es su documentación. Por eso es tan importante una buena documentación de la misma, tanto para el creador, como para el cliente.

Este estado del arte ofrece una estructura bastante lineal, que procedo a recorrer.

Definiciones

Open Data

Actualmente la tendencia a la transparencia de los organismos está pasando a un primer plano. Y una de las maneras de atestiguarlo es la cantidad de información que se publica y es accesible por el público (Cuadrado, 2016). El movimiento Open Data aboga por la libre circulación de los datos con el fin

de incrementar su valor y su usabilidad por la sociedad. Este movimiento se organiza alrededor de tres principios: datos abiertos, participación, y colaboración. Se pueden leer críticas hacia el movimiento por su cercanía a valores neoliberales, por aquellos que abogan que dan más poder a los empoderados, y los peligros sobre la privacidad (Kitchin, 2014). Como expone, las compañías privadas pueden sacar beneficio de la liberación de estos datos para sus intereses comerciales. Y empujados por intereses, también pueden acrecentar las brechas sociales en lugar de acercarnos a una democratización gracias a la liberación información.

Pero no sólo vemos esta tendencia por parte de las compañías privadas, también la vemos llevada a cabo por organismos públicos de hacer accesible toda la información posible sin vulnerar la privacidad. Se puede ver el auge del movimiento Open Data movido por este fin. La tendencia del movimiento Open Data es liberar la información mediante ficheros de datos estructurados, ya que son más sencillos de manipular por programas sin hacerlos excesivamente complicados al programador. Por tanto, más accesibles.

Tim Berners-Lee en febrero del 2009, en una charla TED (“The Next Web of Open, Linked Data”), expresó su frustración por la gran cantidad de información a la que sólo se puede acceder con un determinado conocimiento específico. Internet fue una respuesta al acceso y uso de la información. El acceso libre a la información puede reportar, y reporta, grandes beneficios a la sociedad: desde ayuda en crisis producidas por desastres naturales, para ayuda y rescate, hasta uso en estudios antropológicos y sociales. No sólo es el público general el que se beneficia de este movimiento, sino también la propia administración eliminando barreras artificiales y administrativas en la compartición de datos (Black, A. ,2012).

El mantra del movimiento Open Data es que los científicos compartan todos aquellos datos derivados de su investigación, gobiernos, industrias,..., para poder analizarla y optimizar las tareas. Los avances científicos se verían acelerados si todo el mundo compartiera conocimientos de una manera libre. Pero no todo el mundo está dispuesto a eso. Y lo mismo ocurre con el desarrollo tanto social como industrial, con los que se podrían conseguir grandes avances rápidamente.

El Open Data Handbook (Open Knowledge Foundation 2012) define el Open Data como los datos que se pueden utilizar libremente, reutilizar y redistribuir por cualquiera; sujeto únicamente, a lo sumo, a la obligación de atribuir y compartir por igual (What is open Data, s.f). E intenta clarificar un poco más subrayando los siguientes aspectos: disponibilidad, acceso, reusabilidad y redistribución de una manera no limitada dentro del buen uso de los datos y de su fuente original.

PLOS (Public Library of Science) puso recientemente como requisito que los autores que publicasen en sus artículos científicos publicasen todos los datos necesarios para poder reproducir sus estudios. No fueron los primeros, pero sí los más relevantes (Van Noorden, R. 2014).

En cuanto a los gobiernos y organizaciones civiles, se ha creado una alianza para que estos sean más abiertos y transparentes al ciudadano: el “Open Government Partnership”.

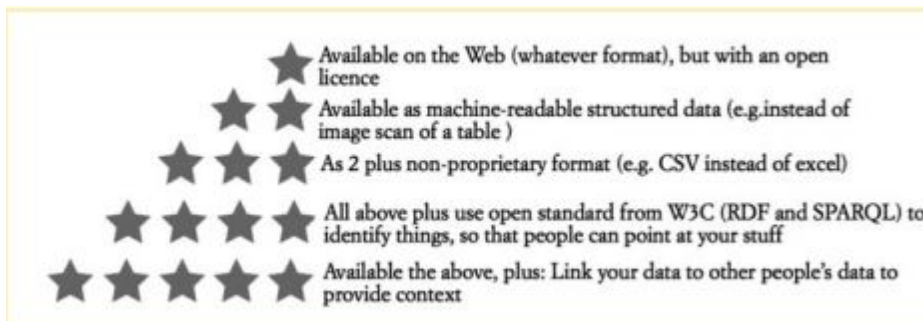
La idea de la interrelación de la información, tal y como la presenta Tim Berners Lee, inventor de internet, es que ésta -la información- se muestre de una manera estructurada, interrelacionada, y capaz

de ser procesada por computadora, tanto en su formato habitual como en una forma en la que se puedan transmitir y ser mostrada a través de la red.

Formato de los datos

La forma de compartir información es importante, pero también lo es cómo está organizada y estructurada. Y cada vez más somos más conscientes de este hecho y tomamos medidas al respecto. Es importante conocer el tamaño, la cantidad, de los datos a los que queremos acceder. Sobretudo en términos de software; para su posterior tratamiento (Cuadrado, 2016).

Los datos son muchas veces publicados en formatos propietarios, por ejemplo xls o pdf. Tim Berners-Lee, en 2010, presentó una clasificación para los formatos Open Data. Esta clasificación está basada en la facilidad de procesamiento y redistribución de los datos en dichos formatos. Esta clasificación otorgaba entre 1 y 5 estrellas, una estrella para aquel formato difícil de procesar (por ejemplo pdf), y cinco estrellas para formatos de datos que permiten de una manera sencilla su reusabilidad y ser enlazados por otras fuentes de datos (Rinciog, O., & Posea, V. 2016)



(Linked Data, s.f)

El formato en el que se presentan los ficheros de datos abiertos es de suma importancia, y cada vez más somos más conscientes de ello. Estos ficheros de datos deben publicar la información en un formato estructurado para facilitar su uso por parte de los lenguajes de programación, de las APIs y programas que hacen uso de éstos (Open Data, s.f). Al facilitar más libertad para la presentación de estos datos se multiplican las opciones para su acceso como el tratamiento, difusión y reutilización. De esta manera, los datos se suelen publicar en: CSV, XLS/XSLX, ODS, JSON, SHP, XML, etc. (García, 2014)

Normalmente el acceso a los datos abiertos se realiza a través de APIs. Las APIs son interfaces para el programador para acceder a estos datos. Como normalmente esta información no va a cambiar mucho, o no va a cambiar en el tiempo, así como igual se trata de información de sólo lectura, los servicios ofrecen esta información en simples ficheros de texto o en hojas de cálculo. XML y JSON son dos de los tipos de fichero más utilizados para este propósito. Existe actualmente un amplio número de herramientas disponibles para leer estos datos, y cada día se desarrollan nuevas. Cuando simplemente se desea descargar estos datos, se suelen utilizar ficheros CSV o TSV. Excel los reconoce sin ningún problema. Excepcionalmente también se usan formatos especializados, por ejemplo MARC para datos bibliográficos. Otra aproximación para publicar estos datos en la red ha sido, y es, la de integrar

directamente los datos estructurados en las páginas web. Empresas como Bing, Google, y Yahoo, a raíz de una iniciativa de Schema.org tienden a unificar criterios en su manera de integrar estos datos estructurados (Stephens, 2011). También nos podemos encontrar con lenguajes o entornos de programación estadísticos, como R, para el mismo propósito. Existen análisis sobre cómo éste y otros se comportan con los tipos de fichero más habituales según el número de datos que contienen y el tiempo de carga que requieren (Cuadrado, 2016).

De entre los tipos de ficheros más utilizados encontramos, y procedo a describir brevemente, los siguientes más comunes:

- **.csv/.tsv:** Los archivos CSV (del inglés comma-separated values) son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas (o punto y coma en donde la coma es el separador decimal: Argentina, España, Brasil...) y las filas por saltos de línea. Los campos que contengan una coma, un salto de línea o una comilla doble deben ser encerrados entre comillas dobles (Valores separados por comas, s.f). Los ficheros TSV son el equivalente a los anteriores, pero en lugar de comas, usa tabulaciones para la separación de los datos.
- **.json:** JSON, acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente (JSON, s.f).
- **.ods:** ODS (del inglés Operational Data Store), es un contenedor de datos activos, es decir, está diseñado para integrar datos de múltiples fuentes con los que realizar operaciones adicionales en los propios datos. A diferencia de un almacén de datos maestro, los datos no vuelven de nuevo a los sistemas operacionales de origen, sino que se preparan para la realización de otras operaciones o para ser incorporados a un almacén de datos (o data warehouse) para ser explotados en la realización de informes (Almacén operacional de los datos, s.f).
- **.shp:** El formato ESRI Shapefile (SHP) es un formato de archivo informático propietario de datos espaciales desarrollado por la compañía ESRI, quien crea y comercializa software para Sistemas de Información Geográfica como Arc/Info o ArcGIS. Originalmente se creó para la utilización con su producto ArcView GIS, pero actualmente se ha convertido en formato estándar de facto para el intercambio de información geográfica entre Sistemas de Información Geográfica por la importancia que los productos ESRI tienen en el mercado SIG y por estar muy bien documentado (Shapefile, s.f).
- **.xls/xlsx:** Formato propietario de las hojas de cálculo Excel. El uso de esta hoja de cálculo está muy difundido. A pesar de ser propietario es de los más extendidos, y aporta mecanismos para un trabajo relativamente cómodo de datos simples facilitando herramientas para poder tratarlos y generar datos complementarios de gran valor en base a éstos.
- **.xml:** siglas en inglés de eXtensible Markup Language ("lenguaje de marcas Extensible"), es un meta-lenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Proviene del lenguaje SGML y

permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información (Silberschatz, A., 2002).

Estado actual de los datos abiertos

El derecho de acceso a la información es un derecho fundamental, y actualmente está reconocido por la legislación internacional, por muchas constituciones, y por más de 80 leyes nacionales de todo mundo. Y es una de las razones por las que cada vez más se tiende a la publicación de datos. Es también conocido como el derecho a saber o libertad de información. Implica el derecho de toda persona a solicitar y recibir información de entidades públicas. A su vez, obliga a las entidades públicas a publicar información sobre sus funciones y gastos de manera proactiva (Coalición Proacceso, 2016).

El fenómeno de los datos abiertos explota y profundiza en las oportunidades sociales y comunitarias que permiten, desde hace una década, los servicios y contenidos digitales abiertos. La falta de una cultura de transparencia en nuestra sociedad, y de rendición de cuentas, la ausencia de una regulación clara hasta hace pocos años, y la imposibilidad de obtener tecnologías apropiadas para acceder a la información, ha cercenado oportunidades para obtener acceso fácil y directo a conjuntos de datos y microdatos de interés público (García, 2014).

Dada la avidez actual de la sociedad por la información para hacer uso de ésta, se está incrementando sobremanera el número de sensores y dispositivos inteligentes capaces de generar información a través de las costumbres diarias. Un ejemplo de esto lo podemos ver en el IoT. También muchos gobiernos, como el de la Unión Europea, el de los Estados Unidos, Japón y Australia han mostrado su interés hacia los datos abiertos. Han visto que de esta manera pueden obtener cuantiosos beneficios económicos, así como una mejor imagen de cara a su ciudadanía.

Pero no sólo está dejando huella el uso de los datos abiertos en el ámbito social, político, o económico. También lo está haciendo en el ámbito científico. La publicación abierta de los datos de investigaciones está favoreciendo la reusabilidad de los datos evitando así la duplicidad de experimentos, y facilitando la recreación de los mismos para comprobar su veracidad. Nielsen ya señaló que los beneficios del Open Data no solamente están en la publicación y disponibilidad de estos datos, sino que es un cambio cultural y de conducta:

“The reinvention of discovery is one of the great changes of our time. To historians looking back a hundred years from now, there will be two eras of science: pre-network science, and networked science. We are living in a time of transition to the second era of science. But it’s going to be a bumpy transition, and there is a possibility it will fail or fall short of its potential.”
(Nielsen, 2011, p.x. 10)

La apertura de los datos almacenados por las administraciones y su puesta a disposición de ciudadanos y empresas está permitiendo el nacimiento de un nuevo modelo de negocio basado en la creación de productos y servicios de valor añadido a partir de esa información pública.

El Observatorio Nacional de las Telecomunicaciones y de la Sociedad de la Información (ONTSI), en 2012, realizó un estudio del sector infomediario en España (Caracterización del sector infomediario en España, s.f). Más de 150 empresas habían desarrollado algún servicio a partir de datos públicos. Pero, sin embargo, esta cifra puede ser aún mayor puesto que existe un mercado paralelo de actividades relacionadas con los datos, que actualmente no se recogen en el estudio y son difíciles de cuantificar. ASEDIE (Asociación multisectorial de la información) estima un negocio en torno a 50000 millones de euros teniendo en cuenta posibles potenciales sinergias entre el movimiento Open Data y la industria de contenidos digitales.

Fuentes de datos abiertos

A raíz de los beneficios que conlleva, en estos últimos años estamos viendo, y también veremos, cómo muchas administraciones de los gobiernos están publicando datos para el beneficio de la sociedad. El problema que se está generando es un cuello de botella, se está empezando a liberar muchos datos pero existen pocas herramientas, o medios, para que una persona sin conocimientos técnicos puede hacer uso de ellos. Por eso, cada vez se desarrollan APIs y herramientas para poder explotarlos y hacer uso de ellos.

Algunas de estas fuentes son:

Autonómicos y locales	Estatales
Aragón (Aragon Open Data, s.f)	Gobierno de España (Datos.gob.es, s.f)
Ayuntamiento de Valencia (Valencia: Datos abiertos, s.f) donde podemos encontrar las siguientes categorías: <ul style="list-style-type: none"> ● Cultura y ocio ● Educación ● Medio ambiente ● Servicios sector público ● Sociedad y bienestar ● Recursos sociales ● Tráfico y transporte ● Urbanismo e infraestructuras ● Vivienda 	Ministerio de industria (Sede-e, s.f)
	Europa
	Unión Europea (Portal de datos abiertos de la Unión Europea, s.f)
	Científicos
	CERN (Cern: Open Data, s.f)
	Empresas del sector de la energía
Barcelona (OpenDataBCN, s.f)	Enel (Enel Open Data, s.f)
Cataluña (Datos abiertos GenCat, s.f)	

Y cada vez más son las empresas o instituciones las que se deciden a publicar datos en su poder para el beneficio tanto propio como ajeno.

Uso del OpenData por parte de empresas privadas

No sólo el sector público se decide a publicar información, el sector privado también produce gran cantidad de datos, no sólo utiliza datos públicos. Aunque muchos de éstos están directamente relacionados con estos a través de diferentes regulaciones. Muchos negocios están adoptando prácticas abiertas y colaborativas. No lo hacen de manera altruista o la filantrópica, sino que lo hacen porque realmente porque ésto puede mejorar sus negocios. JP Morgan Chase, por ejemplo, proporciona información a las autoridades bancarias sobre sus filiales; Walmart trabaja para ofrecer visibilidad en su cadena de suministro a través de retroalimentación en tiempo real, y anónima, de los trabajadores de 279 fábricas en Bangladesh; Armajaro trata de rastrear todo su cacao de Ghana a través de dispositivos portátiles y luego comparte información tanto con sus compradores como con autoridades reguladoras y productores para mayor sostenibilidad y rastreabilidad de su recorrido; la minería de Kalsaka en Burkina Faso hace que sus contratos gubernamentales puedan ser accedidos de una manera abierta para que las comunidades locales puedan comprobar si la compañía cumple con sus compromisos ambientales y laborales (The Next Frontier for Open Data: An Open Private Sector, s.f)

Los siguientes ejemplos, también, recogen información sobre empresas privadas que utilizan los datos abiertos como fuente de información para llevar a cabo su negocio (5 empresas con proyectos basados en Open Data, s.f):

- [IDC Consultin](#): consultoría de energía sostenible que utiliza OS OpenData para identificar las ubicaciones óptimas y el terreno para las obras de energía renovables como parques eólicos y solares. La ubicación es fundamental para el éxito de las instalaciones de energía; integran los datos de SO IDC e identifican con mayor rapidez y precisión posibles lugares.
- [CH2M HILL](#): empresa de ingeniería que utiliza OS OpenData para crear una herramienta automatizada para dibujar una red de autobuses multimodal desde cualquier parte de Gran Bretaña. Consigue reducir el tiempo necesario para analizar toda una red de autobuses en un 90%.
- [Cítricos Analytics](#): empresa de análisis de marketing que utiliza el OS OpenData para mejorar sus datos, dando la posibilidad de visualizar los clientes en un mapa y poder ser utilizado como base de grupos de investigación cualitativa.
- [T4 Media](#): empresa de publicidad que utiliza OS OpenData para automatizar el proceso de identificación de espacios de publicidad exterior más adecuados y permitir el análisis de múltiples puntos de sus escritorios.
- [Rubicon Patrimonio Services Ltd](#): empresa de arqueología que utiliza OS OpenData para habilitar la asignación de fondos en su equipo de prospección y ofrecer verificación en tiempo real. Así, recopilan de forma correcta y accesible.

Podemos encontrar más casos en OpenDataHandBook (OpenDtaHandbool: Value stories, s.f)

También merece la pena mencionar el caso paradigmático de Celera Genomics contra el Proyecto del Genoma Humano para secuenciar de forma completa el genoma humano, donde se enfrentó el interés

privado con el interés general de la sociedad. Y se puso en evidencia la ética y los intereses científico privados (Open data en el sector privado, 2010).

Open Data Vs. Big Data

Una vez expuestos varios ejemplos, nos cabríamos preguntar cuál es la diferencia entre Open Data y Big Data. Según la definición de diccionario Big Data es un conjunto muy grande y complejo de datos para ser procesados por un sistema gestor tradicional de base de datos (Big data, s.f).

Cada días generamos 2,5 quintillones de bytes de datos. Y alrededor del 90% de los datos recogidos de información han sido generados en los últimos dos años. Estos datos provienen de una gran diversidad de fuentes, desde sensores que recogen información ambiental, hasta la información recabada de los posts de la gente en redes sociales, pasando por la información audiovisual, transacciones electrónicas, redes móviles, etc (Bringing big data to the enterprise, s.f). Otras personas, como Stephane Hamel comentó en agosto del 2012 en *Big Data - What It Means For The Digital Analyst*, que el término Big Data se asigna a aquellas colecciones de datos no susceptibles de ser representadas, o manipuladas, con Excel. Tenemos, por tanto, una definición bastante amplia de Big Data. Pero Big Data no quiere decir Open Oata. Los datos del Big Data no tienen porqué ser abiertos. La definición ofrecida por Gartner (What Is Big Data?, 2016), que dijo que Big Data representa una gran cantidad de información, accesible a gran velocidad y/o de gran valor añadido que requieren un procesado adecuado y con gran intencionalidad y especificidad, que ayuden a tomar decisiones y a los procesos de automatización de la información de una manera eficiente, también en términos de coste. Lo expresa más en forma de cómo procesamos esa información, con la finalidad establecida. Esta definición no tiene porqué alejarse tanto de la definición de Open Data.

Problemática y posibles soluciones

Problemas y soluciones de los datos abiertos

Uno de los problemas que muchas veces se tiene para hacer uso de datos es por la problemática de las licencias de uso y explotación de los mismos. Mucha información se puede encontrar online, pero muchas veces no se utiliza (Skelton, 2011). Parece axiomático que quien proporciona datos Open Access lo haga sin ningún tipo de barreras, indicándolo en las propias webs desde donde son accesibles. Y muchos lo hacen. La manera más adecuada es someter a esos datos a algún tipo de licencia. La Creative Commons - Acknowledgement (CC-BY) cumple con la intencionalidad deseada, y es la más común cuando se realiza esta práctica. Sin embargo, nos encontramos con varias webs que publicitan sus datos como Open Access pero que limitan el uso de los datos que ofrecen, la mayoría de las veces a un uso no comercial de los mismos a través de una licencia con este propósito como es la Creative Common non-Commercial (CC-NC). Pero si analizamos este tipo de licencias, observamos que fueron creadas orientadas hacia entornos de música, literatura, imagen,... , y nos se ajustan a

ámbitos escolares,científicos,... , en los que se pretende usar esta información Open Data. (Murray-Rust, 2008)

En cuanto a las licencias en el contexto de la publicación de contenidos, el modelo que mejor responde a la necesidad de difundir los datos abiertos al mayor número posible de ciudadanos -tal y como se ha mencionado anteriormente- son las licencias Creative Commons¹, que nos permiten gestionar los derechos relacionados con su difusión: el reconocimiento de la autoría; el uso comercial y la producción de obras derivadas.

Por otra parte, el Open Data Commons promueve el uso de tres tipos de licencias básicas para los datos y bases de datos, que responden en mayor o menor medida a los requisitos mencionados anteriormente sobre su difusión, utilización y reutilización (Licenses, s.f). Todas estas licencias permiten la copia o distribución de los datos, la producción de obras basadas en los datos y la adaptación o modificación de las bases de datos, sin restricciones; la única obligación es la de reconocer el autor de la información generada, y con el objetivo de promover el carácter abierto de los datos originales, que sólo puedan ser utilizados cuando los productos o adaptaciones se publiquen, divulguen, sea bajo la misma licencia.

Retos de los datos abiertos

De entre todas las razones que existen para abrir los datos, hacerlos públicos, nos encontramos las siguientes: (1) recogemos datos públicos con recursos públicos, por lo que sería re recibo mostrar cuál ha sido el resultado de esa inversión, (2) es beneficioso para el país (Informe PIRA, 2000), (3) lo recomienda el G8, G20, ONU, (4) los datos ya no son negocio (8 % presupuesto CNIG), (5) permite aprovechar la IGV, (6) nos permite competir en igualdad de condiciones con empresas ya cimentadas, (7) los datos de referencia deben ser abiertos y (8) mejora nuestra imagen y es una demanda social (Hayashi, 2016)(XI Jornadas internacionales gvSIG, 2015).

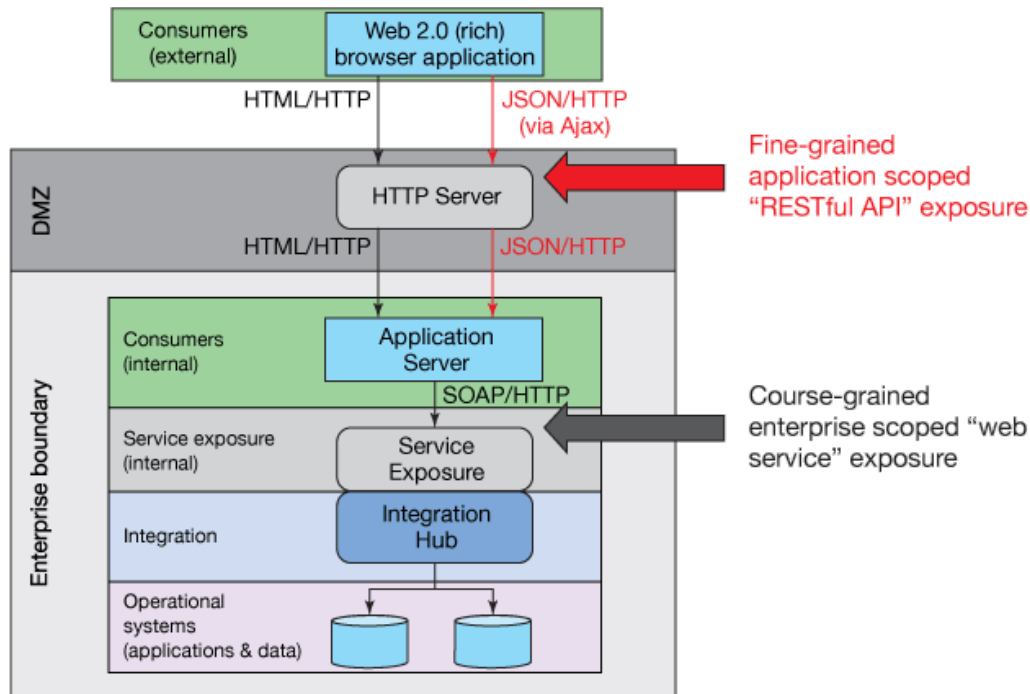
Las instituciones públicas han recibido bien las bases de este movimiento, y nos les cuesta ver los beneficios que aporta la liberación de los datos no sensibles. Muchas veces estas instituciones no tienen los fondos necesarios para poder explotar estos datos y que la sociedad los pueda aprovechar. Por lo que no ven problema en ofrecerlos y que parte de la comunidad participe de esto desarrollando medios que beneficien al conjunto social gracias a esta información. Más difícil es, a priori, que el sector privado pueda ver beneficios con esta táctica. Sin embargo,cada vez más son más conscientes de los beneficios que les pueden aportar (Say, M. 20 diciembre de 2013).

WebAPIs

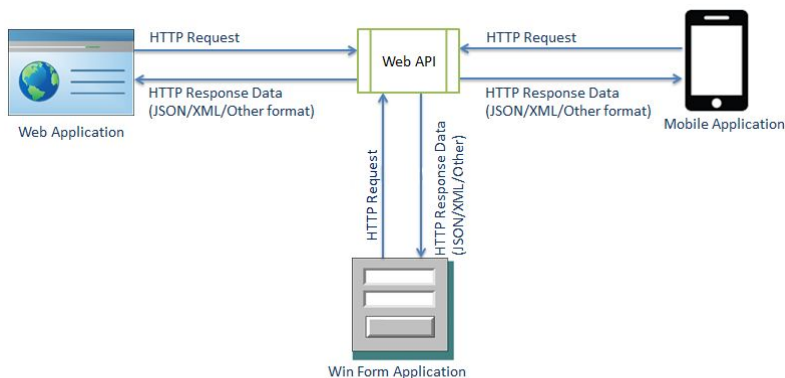
Una API (Application Programming Interface) es una interfaz para que un cliente interactúe con un sistema servidor. El desarrollador sólo se preocupará de cómo interactuar con la API, y no de cómo

¹ <http://es.creativecommons.org/blog/licencias/>

ésta está implementada. La descripción de estas APIs, por lo tanto, es algo deseado y recomendado. Nos ayudan a comunicar las funcionalidades de ésta, y facilitan su adopción.



Hoy en día ya casi no podemos imaginar una aplicación Web, móvil, página, etc, que no aproveche algún tipo de recurso remoto. En el desarrollo de funciones y programas se está desarrollando una manera a través de la red para contactar con estos recursos. Se hace a través de APIs. De las aproximaciones llevadas a cabo, parece que el estándar de facto que se va a establecer es el de usar el protocolo HTTP. Éste se puede usar para enviar y para recibir información a través de la red. Habrá, por lo tanto, dispositivos con los que nos comunicaremos y utilizaremos estas Web APIs para obtener y/o procesar información. Una Web API, por lo tanto, es una interfaz de un servicio web a la que se puede acceder mediante las acciones definidas por el protocolo HTTP. Ofrecen un métodos para acceder a bases de datos de servicios que se ofertan en Internet, son una manera de atacar a una base de datos a la que atacan servicios Web.



La descripción y documentación de Web APIs son la base para los programadores en los desarrollos de los próximos años (Young, 2016).

Estas Web API se hacen públicas para que los programadores puedan hacer uso de ellas. También para aprender de ellas, tanto de su funcionamiento, como del negocio detrás. Por eso, se suelen publicar y promocionar en repositorios públicos.

Repositorios públicos de WebAPIs

El éxito de los servicios Web, y de las API para hacer uso de ellos, lo podemos observar analizando su crecimiento, que es continuo y sostenido, de estos directorios de WebAPIs.

ProgrammableWeb, por ejemplo, es un muestra del éxito que tienen los servicios web y las API entre los desarrolladores. En su web [ProgrammableWeb](#)² nos encontramos con un gran repositorio de web APIs. Las podemos encontrar clasificadas por categorías. En ella se enlaza con las distintas webs, o negocios, que ofrecen dichas APIs para obtener/procesar la información que ellas mismas ofrecen. Nos encontramos con una variedad que va desde [Google Maps](#), hasta [Wikis](#), pasando por APIs de servicios de pago como [PayPal](#).

En esta web nos encontramos información básica sobre cada API ofrecida. Nos indica qué SDK podemos utilizar para desarrollar las herramientas que harán uso de estas APIs. Al mismo tiempo nos indica, también, cómo utilizar la API en cuestión, Y nos enlaza a entradas o artículos que explican cómo trabajar con ella, y cómo beneficiarnos de ella de una manera eficiente dándonos indicaciones o consejos para este fin. Y como una manera de iniciación, o de consulta, nos da códigos de ejemplo de uso de las mismas e indica qué librerías podemos utilizar para ayudarnos a la hora de trabajar con ellas. Nos da también información de contacto de los desarrolladores y seguidores de estas APIs para poder hacerles cualquier tipo de consulta. De las APIs más populares hay más información que de las APIs menos populares. Y esto depende del número de usuarios de los que disponga. De las primeras sí que ofrecen, de manera más habitual, códigos de ejemplo y ejemplos de uso. De las segundas habrá que dirigirse a la página del servicio para encontrar información de la API y de la manera de hacer uso de ella.

Si cogemos una API al azar, por ejemplo la de Stripe, podemos encontrar su [localización](#), [la localización de la documentación de su API](#), a qué categoría pertenece, categorías secundarias con la que está relacionada, si soporta SSL o no, [su dirección de Twitter](#), la [dirección de contacto](#), su [dirección de soporte para su desarrollo](#), [los términos de servicio](#), si es propietaria o no, su ámbito de uso, si es propia de un dispositivo o multidispositivo, el tipo de arquitectura de la API, los formatos que soporta tanto para realizar peticiones como para recibir respuestas, si es oficial o no, si es una API que soporta hipermedia, y si requiere permisos para acceder a ella.

Para cada una de las APIs también disponen de secciones que ofrecen un sumario de cada una de ellas, SDKs (herramientas para poder utilizar estas APIs), información sobre cómo usarlas, códigos de

² www.programmableweb.com

ejemplo, librerías, un listado de desarrolladores, un listado de gente que hace un seguimiento de la API en cuestión, y una sección para poder dejar comentarios sobre la API.

Los beneficios que se obtienen de este tipo de repositorios es muy amplio. Nos encontramos ante un compendio de APIs donde se especifica cómo trabajar con ellas, y códigos de ejemplo. Ésto facilita mucho la labor, y la agiliza para poder usarlas de una manera eficaz en el menor tiempo posible.

Podemos encontrar más repositorios donde localizar APIs, tanto libres como privativas. Por ejemplo Mashape (MarketPlace, s.f). Aquí las podemos encontrar también clasificadas por categorías, y podemos clasificarlas, también, utilizando distintos criterios (popularidad, relevancia y antigüedad).

RESTFul WebAPIs

De entre todas las WebAPIs que podemos encontrar, me voy a centrar, en particular, en las RESTFul WebAPIs. Éstas, son APIs que siguen una filosofía/metodología REST. REST es un estilo, una guía, de cómo utilizar HTTP para el desarrollar APIs escalables y robustas. Para gestionar gran variedad de recursos (como podemos ver en la figura), las RESTFul WebAPIs deben basarse en el conjunto estándar de operaciones (Post, Get, Put, Delete, Options, Head) que define HTTP (Rodríguez, 2016) (Havens, A. 2016):

- Post, debe ser utilizado para crear nuevos recursos dentro de una colección.

```
POST http://ejemplo.com/objetos
```

Data:

```
name = Objeto1
```

- Get, debe utilizarse para recuperar una representación de un recurso.

```
GET http://ejemplo.com/objetos/123
```

- Put, debe utilizarse para actualizar o crear recursos.

```
PUT http://ejemplo.com/objetos/123
```

Data:

```
name = New name
```

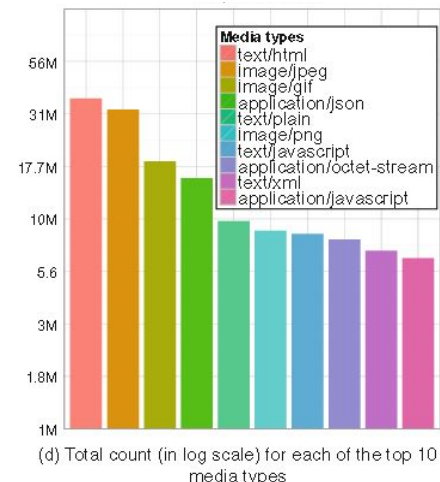
```
color = azul
```

- Delete, debe utilizarse para eliminar un recurso de su padre.

```
DELETE http://ejemplo.com/objetos/123
```

- Options, debe utilizarse para recuperar las interacciones disponibles de un recurso.

- Head, debe ser utilizado para recuperar los metadatos del estado actual de un recurso.



Tanto desarrolladores como no desarrolladores necesitan ser capaces de elegir la API más adecuada a sus necesidades de una manera rápida. Cuanto más rápido y con menores costes mejor. Que esté bien pensada y diseñada será un factor determinante a la hora de tomar esta decisión. Una WebAPI bien

diseñada considera los siguientes aspectos y es coherente con sus necesidades: (1) el modelado de los datos, (2) la identificación de las fuentes de datos (URIs), (3) su representación, (4) la definición de sus operaciones (HTTP), (5) la localización de los datos (Rodríguez, 2016). La que mejor se ajuste, por tanto, a las necesidades del programador será la que más probabilidades tenga para ser seleccionada y trabajar con ella.

Lenguajes para crear RESTful WebAPIs

En la selección de un API es también importante utilizar el lenguaje en el que está escrita como criterio de selección. Normalmente, ya que van a usar también el conjunto de librerías asociadas a estos lenguajes para facilitar así su programación, se eligen los lenguajes más populares para esta tarea (los que ofrecen más y mejores librerías). En la página mencionada anteriormente, ProgrammableWeb (ProgrammableWeb, 2013), en el apartado de repositorio, hacen una clasificación de uso según las librerías asociadas a cada lenguaje:

- PHP (47)
- Python (46) - Django, Pyramid, Flask (micro-framework), Swagger
- Ruby (44)
- .NET / C# (38)
- Java (37)
- Perl (26)
- ColdFusion (12)
- Node / JavaScript (8)
- ActionScript (6)

Uso de filosofía y servicios REST desde varios lenguajes

La elección de un lenguaje para crear RESTful WebAPIs para atacar la información que podemos encontrar va a depender principalmente de la soltura y preferencia de un lenguaje sobre otro. Tanto del programador/es, como de la compañía/institución. Actualmente la mayoría de los lenguajes dan soporte y ofrecen librerías para poder trabajar cómodamente con esta metodología. También vendrá condicionado por dónde se quiera/requiera utilizar la aplicación de marcos, de los datos con los que se vaya a tratar. Como recomendación, adaptarse a lo que tenemos y con los que vamos a trabajar. De entre los tipos de datos que nos encontramos, JSON es uno de los estándares más utilizados para las API REST, si no el que más; si nuestros datos van a ser de tipo JSON lo más recomendable podría ser usar NodeJS y JavaScript, puesto que trabajan de forma nativa con este tipo de datos. Aunque tampoco se puede considerar relevante ya que la mayoría de los lenguajes (y entornos de trabajo) tienen muy buenas bibliotecas que ayudan en la creación de las API. Habría que analizar la naturaleza y uso que se le va a dar a los datos -a quien van dirigidos y con qué propósito-.

OData

Del término OpenData se tiene por primera vez referencia en 1995. Y cada día que pasa se está haciendo más común. Surgió con la idea, y la conciencia, del deber compartir datos de investigaciones, de la compartición de saber. Posteriormente hizo sinergia este movimiento con el de software libre. Y ya en 2007 se empezó a materializar. Así, muchos gobiernos, viendo su beneficio, empezaron a apoyar esta filosofía. El movimiento Open Data está llegando a tener mucha importancia, tanto política como económica. Muchas veces resulta difícil justificar la liberación de determinados datos, pero en la mayoría de las ocasiones, si no en todas, se puede sacar provecho de ellos.

Ante el gran abanico de posibilidades que hay Microsoft se decidió también a impulsar la iniciativa para abrir datos al público mediante una evolución de lo que era anteriormente conocido como ADO.NET Data Services (proyecto Astoria). Apareció en el 2010, y desde entonces se han incrementado los productos, webs y tecnologías que exponen sus datos utilizando este protocolo.

OData está construido sobre Atom Publishing Protocol o AtomPub especificado en RFD 5023 quien a su vez está construido sobre Atom Syndication Format o Atom especificado en RFC4287. Y se basa en: (1) la creación de una forma uniforme de representación de datos estructurados (JSON, Atom), (2) convenciones URL y, (3) la filosofía REST.

Entre las empresas tiene bastante acogimiento. Entre ellas podemos encontrar algunas destacadas como eBay, Facebook Insight, IBM WebSphere, Netflix, twitpic, GeoREST, Nuget, Windows Live o DBpedia entre otras en el sector privado y Open Government Data Initiative (OGDI), The City of Edmonton Open Data Catalogue o Public Transit Data Community, entre otras.

Con OpenData estas empresas permiten que otras colaboren con ellas a través del uso de sus datos. No sólo eso, también sientan las bases de una política de transparencia que permite un mayor conocimiento de sus productos, valores, funciones que desarrollan, estructura organizativa y posibles planes de futuro.

Open API Initiative

Wordnik.com es un diccionario de Inglés online, una organización sin ánimo de lucro, y su función es la de dar soporte a la lengua analizando los recursos de internet. Fué aquí donde se gestó Swagger, una especificación para trabajar con los servicios web con filosofía REST. En 2015 SmartBear creó una nueva organización, apoyada por grandes compañías como IBM, Google, Microsoft, ..., que se llamaría Open API Initiative. En enero de 2016 la especificación de Swagger pasó a llamarse Open API Initiative. Su uso permite generar automáticamente la documentación de modelos, métodos y parámetros. En principio es independiente del lenguaje de programación utilizado. Más adelante se muestra un cuadro comparativo de Swagger, APIBlueprint y RAML. Estas últimas pertenecen también a este grupo y adoptan también esta filosofía.

El compañero natural del Open Data son las Open API. Éstas son APIs que, al contrario que las API privativas, están abiertas a los desarrolladores. Éstos pueden hacer uso de ellas para acceder a los datos que ofrecen, lo que aporta gran valor tanto a la organización (como accesibilidad, transparencia,...) como al producto que ofrecen. Hay que tener, sin embargo, cuidado con esos datos. Se tendrá que analizar la naturaleza de los mismos. También, cuidado con ofrecer una API muy específica, centrada a un conjunto muy específico de destino. Si este es el caso, estaremos acotando el alcance de la misma y, por tanto, de potenciales usuarios que podrían ampliar el abanico de destinatarios potenciales a través de su uso. Si tenemos esto como consideración, el desarrollo de estas APIs abiertas potencia y da valor al producto de la compañía que la tiene, y es una manera, con un bajo coste asociado, de complementar su valor y poder explorar nuevas posibilidades de negocio fruto de lo que ellos mismos disponen -información-.

Ejemplos

Debido a las razones expuestas anteriormente, hay muchas organizaciones que abren sus APIs para que los programadores las podamos usar. Y la tendencia actual es creciente. Podemos encontrar muchos repositorios en Internet que indican las especificaciones, dan ejemplos, muestran sus particularidades y dicen dónde las podemos encontrar. Entre otros, destacan: ProgrammableWeb³ (ya mencionada anteriormente), PublicAPIs⁴, etc. El abanico de estas APIs es bastante amplio, muchos negocios y organizaciones se están sumando a esta tendencia abriendo las posibilidades que les brindan este tipo de acciones. Por ejemplo, en el campo de los pagos online encontramos, entre muchas otras: Amazon FPS (Flexible Payments Service)⁵, PayPal Invoicing⁶, Payment Express⁷, PAYMILL GmbH⁸, BlockChain⁹,... Por otro lado, por ejemplo, relacionadas con el ámbito de publicaciones científicas: Elsevier¹⁰, Springer¹¹, Altmetric¹², ... Cada una de ellas ofrece una descripción detallada de los servicios que ofrece, recomendaciones de uso, y cómo usarla (ya que su propósito es que sea utilizada por el mayor número de servicios/sitios posibles).

E igual que a nosotros nos conviene que las API que queremos utilizar estén bien pensadas y documentadas, a nosotros también nos conviene eso de las propias. Cuanto mejor sea su documentación y documentación, más posibilidades tendremos de que otros programadores las utilicen.

³ <https://www.programmableweb.com>

⁴ <https://www.publicapis.com/>

⁵ https://amazonpayments.s3.amazonaws.com/FPS_ASP_Guides/ASP_Getting_Started_Guide.pdf

⁶ <https://www.x.com/developers/paypal/products/invoicing>

⁷ <https://code.google.com/archive/p/payment-express/>

⁸ <https://market.mashape.com/paymill/paymill-gmbh>

⁹ https://blockchain.info/api/api_receive

¹⁰ <http://dev.elsevier.com/>

¹¹ <https://dev.springer.com/>

¹² <http://api.altmetric.com/>

Manipulación, documentación y especificación

Para el desarrollo y manipulación de las RESTful WebAPIs nos apoyamos en los métodos que nos proporciona HTTP, y ya mencionados anteriormente: con GET recuperamos una lista o un solo elemento de los datos, con POST se envían al servidor, con PUT se actualiza una fila de datos existente, y con DELETE se pide eliminar una fila de datos. Estos verbos se utilizan para asignar a un método de estas Web APIs. Con la filosofía REST también se pueden manipular datos. Quizás la manera más conocida para este propósito sea CRUD (*Create, Read, Update y Delete*) (Dewailly, L. 2015). Con CRUD los datos se pueden considerar como entidades pasivas, con REST operamos sobre la representación de esos datos. No hace falta tampoco disponer de URLs distintas para cada tipo de operación, siguiendo REST no nos ceñimos al uso de POST para borrar, insertar y actualizar. REST es mucho más que CRUD. Los recursos no son sólo una entrada a los datos, y las operaciones hacen más que sólo actualizar registros.

La mayoría de los lenguajes (y frameworks) tienen bibliotecas para creación de APIs (Rodríguez, 2016). La gran mayoría de colecciones de Web API presentan mucha idiosincrasia, por eso es muy importante aportar una buena descripción y documentación de las mismas. Por este motivo, es de gran importancia saber escoger las herramientas más adecuadas para trabajar con ellas.

Herramientas para crear RESTful WebAPIs

La elección de herramientas para crear RESTful WebAPIs vendrá determinada el lenguaje que queramos utilizar para desarrollarla. Y éste, según el tipo de datos con el que se quiera trabajar, con lo cómodo que se sienta el programador con ese lenguaje, o con la compatibilidad del resto de productos o soluciones que se ofrezcan, y se requiera cierto tipo de integración.

Paso a mostrar un análisis atendiendo al lenguaje con el que se quiera trabajar:

- Si se escoge la plataforma .NET, con ASP.NET tendremos soporte para construir WebAPIs siguiendo el MVC. Es la más usada en entornos de desarrollo Windows. Como es un framework que se ejecuta sobre una plataforma virtualizada, da la posibilidad para elegir el lenguaje que se quiera utilizar.
- Si se elige Java, éste tiene un ecosistema muy grande, con mucha documentación, librerías, y entornos de desarrollo -que también pueden estar especializados-.
- Python es Open Source, por lo que es también bien acogido. De entre sus frameworks destacan Pyramid y Django, que también están más enfocados a aplicaciones grandes. También aportan herramientas para Bootstrap. Pyramid es más flexible, mientras que Django integra todo de una manera más directa. Django incluye un ORM. SQLAlchemy, como alternativa a Django, también es muy popular. En micro frameworks, también Restlet, Jersey, etc. (Brown, 2016).
- En cuanto a Ruby, no escala muy bien cuando hay un número muy grande de peticiones al servidor. Y pese a disponer de muchas librerías sobre las que apoyarnos, se hace más difícil de

mantener a medida que va creciendo. Aquí podemos trabajar con Rails, Grape o Sinatra, entre otros.

- PHP es uno de los más usados. También es de los más veteranos. Tiene menos restricciones a la hora de escribir, lo que lo hace más flexible -menos estricto que otros lenguajes-. Para PHP destacar Slim y Epiphany entre otros.
- Node.js es uno de los más recientes, desarrollado a partir de javascript, y por eso es más difícil encontrar recursos en comparación con otros lenguajes. Aquí destacar las opciones de Restify, Express y Hapi.

Herramientas para documentar RESTful WebAPIs

La documentación es clave para el desarrollo, y difusión de nuestra API. Para la especificación y documentación estas API nos encontramos, principalmente, con: OpenAPI Specification (Swagger), RAML y API Blueprint; que se utilizan para diseñar, describir y documentar las WebAPIs.

Swagger UI funciona bastante bien a la hora de describir y visualizar los servicios web RESTful. Genera una pequeña página web que documenta la API y permite hacer consultas de prueba utilizando JavaScript (Karzyński, 2016). Swagger se basa en YAML o JSON, y es una de las más populares. La especificación es compatible con JSON Schema. Viene con herramientas de código abierto para generar documentación y simular servidores para hacer pruebas. Sin embargo, su interfaz de usuario parece tener mucho margen de mejora. Swagger se centra, principalmente, en la difusión y sincronización de las actualizaciones del workflow. Tanto el cliente como el servidor se generan y documentan de manera simultánea, por lo que se actualizan al mismo tiempo. La codificación del servidor está estrechamente relacionada con su documentación, lo que permite que ésta se genere de una manera muy integrada. Se hace una especificación de abajo a arriba.

RAML es otra opción. No sólo está pensada para la filosofía REST, también sirve para arquitecturas SOAP y RPC. Parece tener una curva de aprendizaje menos pronunciada si las comparamos con API Blueprint y Swagger. Igual que ésta, está muy relacionada con YAML. Puede usarse también como planificador a largo plazo. Y hace una especificación de arriba a abajo. A diferencia también de Swagger, describe el funcionamiento de las funciones a través de recursos anidados, de esta manera está más jerarquizado.

API Blueprint, a diferencia de las anteriores, se apoya más en lenguajes de marcas. Proclama que está más orientado a la documentación, y parece ser así. Hace una especificación de arriba a abajo. Lo que Swagger llama "operaciones", API Blueprint llama "acciones"; también necesita de código de servidor de terceros, cosa que los anteriores no sucede. API Blueprint se centra, principalmente, en C++ con implementaciones Node.js y C#. Sin embargo, uno de los mayores inconvenientes que tiene, es la falta de herramientas para crear y manipular código. (Sandoval, 2015)(Mazaheri, M. 2015).

Comparando las 3 se pueden sacar las siguientes conclusiones según los siguientes puntos:

- Soporte de la comunidad: Swagger es la que tiene la comunidad más activa, la que más soporte por seguidores tiene y la que ofrece más recursos.

- Apoyo de la industria: Tanto Sagger como RAML tiene mucho apoyo detrás.
- Facilidad de uso: APIBlueprint, RAML y Swagger, ofrecen una información a través de sus páginas que es muy útil y facilita el uso de las mismas. Funcionalidad: Aquí destaca RAML.
- Edición: Aquí RAML también. Tanto RAML como Swagger proporcionan editores intuitivos e incorporan una vista previa que permite ver exactamente el resultado final. Sin embargo RAML nos permite la reutilización de código, y construir un diseño de manera gráfica.
- Inicio: La página de API Blueprint es, quizás, más intuitiva. Pero comenzar a escribir cuesta un poco más que con el resto, su curva es más pronunciada al principio en este aspecto. Swagger proporciona todo a través de GitHub, y no es tan intuitivo. RAML parece proporcionar, hasta la fecha, mejores recursos para iniciarse con él, y sin necesitar tener un nivel avanzado.
- Quizás, a la hora de escribir APIs, Swagger sea más cómodo. Es capaz de soportar la mayoría de los lenguajes compatibles para el desarrollo del servidor.
- En cuanto al terminal, tanto el de RAML, como el de Swagger, son fáciles de usar, leer e interpretar. La de RAML parece mostrarse más práctica e intuitiva, en lugar de crear secciones diferentes para cada acción sólo hace falta coger la acción HTTP correspondiente.
- Para generar la documentación, API Blueprint ofrece más opciones y su presentación en una página ofrece más opciones en la presentación. RAML está más limitado en lenguajes que soporta para generar documentación en múltiples páginas.
- Y en cuanto a lenguajes Swagger es el más polivalente.

Paso a mostrar tabla comparativa (Hamza Ed-Douib, 2016):

Información básica			
	Api-Blueprint	RAML	Swagger
Formato	Markdown	YAML	JSON
Licencia	MIT	ASL 2.0/TM	ASL 2.0
Version	1A9	1.0	2.0
Tipo de diseño	Top-down	Top-down	Top-down. Bottom-up a los frameworks con soporte
Soporte	Apiari	Mulesoft, SOA, AngularJS, Intuit, Box, PayPal, & Programmable Web	3Scale, Apigee, Reverb
Grupo de trabajo	No	Sí	Sí
Modelado REST			
	Api-Blueprint	RAML	Swagger
Métodos/Acciones	X ("acción")	X ("método")	X ("operación")
Query Parameters	Sí	Sí	Sí
Path/URL Parameters	Sí	Sí	Sí
Header Parameters	Sí	Sí	Sí
Representations (status codes, mime-types)	Sí	Sí	Sí
Documentación	Sí	Sí	Sí

Autenticación		Basic, Digest, Oauth, 1&2	Basic, API-Key, OAuth 2
Representación de los metadatos	inline	Inline y externa	JSON Schema
Recursos anidados	Sí	Sí	
Polimorfismo/herencia	Resource models	Traits, resource types	
Inclusión de ficheros		Sí	
Metadatos Versión API		Sí	Sí
Representación de ejemplos	Sí	Sí	
Herramientas			
	Api-Blueprint	RAML	Swagger
Authoring	apiary.io	API-Designer	Swagger Editor
Pruebas Ad-hoc	apiary.io	API-Designer	Swagger-UI
API Builder	Node & .Net bindings	MuleStudio APIKit (Visual, arrastrar y soltar), Osprey (Node)	Herramientas para los lenguajes populares
Generador de documentación	sólo una página multipágina	sólo una página multipágina (pero sólo PHP)	Sólo una página
Entorno de prueba	Sí	Sí	Por terceros
Código servidor	por terceros	Java	Java, scala, ruby, node.js
Lenguajes	Node, .Net	JS, Java, Node, PHP, Python, Ruby	Clojure, Go, JS, Java, .Net, Node, PHP, Python, Ruby, Scala
Validación	Sí	Sí (Java)	Sí
Parsing	C++ (nodejs, c#)	Java, js	Java, sj

Herramientas para probar RESTFul WebAPIs

Otro de los aspectos más interesantes es el de poder probar su funcionamiento. Hay muchas herramientas para probar APIs. Entre otros, destacar:

- SoapUI¹³: SoapUI ofrece un entorno de prueba para las REST APIs, soporta WSDL (Lenguaje de descripción de servicios web), y la refactorización de los tests. Es uno de los más usados.
- Mockable.io¹⁴: Mockable.io es un servicio online que permite definir y probar las RESTFul APIs. Y permite importar las definiciones desde Swagger.
- Runscope¹⁵: Runscope es un servicio web que recoge herramientas para comprobar que nuestra API se está ejecutando adecuadamente, comprobar que devuelve datos de una manera correcta, y que se puede hacer un debug sobre ella. También se pueden importar definiciones desde Swagger.

¹³ <https://www.soapui.org/>

¹⁴ <https://www.mockable.io/>

¹⁵ <https://www.runscope.com/>

- JsonStub¹⁶: JsonStub es otra interfaz web que permite simular, sin muchos esfuerzos, servidores que devuelven información de manera estática.
- Restlet Studio¹⁷: Restlet Studio es capaz de describir las APIs de una manera más visual. Es compatible tanto con Swagger como con RAML.
- Apiary.io¹⁸: Apiary.io trabaja con API Blueprint, es capaz de generar un conjunto de pruebas que permite trabajar con ella antes de estar totalmente implementada.
- Postman¹⁹: Postman es un plugin de Chrome el cual también permite construir, documentar y probar las API de una manera rápida.
- Para Firefox tenemos RESTClient²⁰ con características similares

Conclusiones

El uso de los datos abiertos, afortunadamente, se está extendiendo. Si bien antes éramos muy celosos con cualquier tipo de información, cada vez más ésta se hace pública. Haciendo esto conseguimos muchos beneficios, desde dar una mejor imagen de nuestra organización, hasta colaborar con aquellos a la que les pueda beneficiar la información que se publica, pudiendo nosotros también conseguir beneficios indirectos de los resultados que obtengan otros de nuestros datos.

A raíz de este auge, y de la tendencia generada, se hace conveniente conocer las posibilidades actuales para hacer pública distinta información y ofrecer servicios gracias a ésta. Para ello se hace interesante conocer distintos ejemplos de empresas u organizaciones que podamos utilizar de referencia. También qué especificaciones y herramientas tenemos para documentar, crear y nuestras propias WebAPIs. Hago un repaso de los lenguajes más utilizados en la creación de estas APIs, siendo los más comunes Python, PHP, Ruby,... entre otros. En este estudio pretendo mostrar cómo se encuentra actualmente el panorama de posibilidades para realizar estas acciones y dar estos servicios que, como he expuesto, pueden ser muy útiles y van en consonancia con la tendencia actual ya que cada vez más se hacen más evidentes las ventajas que aporta esta tendencia. Y para conseguirlo, comparo tres posibles especificaciones para documentar estas APIs, que además nos sirven de base para la creación de estas mismas. La elección de ésta dependerá también de distintos factores que se han analizado.

Bibliografía

Almacén operacional de los datos. En Wikipedia. Recuperado el 23 de octubre de 2016 desde https://es.wikipedia.org/wiki/Almacén_operacional_de_los_datos
Aragon Open Data. Recuperado el 3 de octubre de 2016 desde <http://opendata.aragon.es/>

¹⁶ <https://jsonstub.com/>

¹⁷ <https://studio.restlet.com/>

¹⁸ <https://apiary.io/>

¹⁹ <https://www.getpostman.com/>

²⁰ <http://restclient.net/>

Big Data. Gartner. Recuperado el 4 de octubre de 2016 desde <http://www.gartner.com/it-glossary/big-data/>

Big data. En Merriam Webster. Recuperado el 31 de octubre de 2016 desde [http://www.merriam-webster.com/dictionary/big data](http://www.merriam-webster.com/dictionary/big%20data)

Black, A. (2012). Open Data Movement. PM.Public Management, 94(6), 22-23. Recuperado desde <http://search.proquest.com/docview/1030964169?accountid=14777>

Bringing big data to the enterprise. Recuperado el 28 de diciembre de 2016 desde <https://www-01.ibm.com/software/data/bigdata/what-is-big-data.html>

Brown, Ryan. Django vs Flask vs Pyramid: Choosing a Python Web Framework. Recuperado el 31 de diciembre de 2016, desde <https://www.airpair.com/python/posts/django-flask-pyramid>

Caracterización del sector infomediario en España. Recuperado el 10 de noviembre de 2016 desde http://www.ontsi.red.es/ontsi/sites/ontsi/files/pr_estudio_sector_infomediario_2012.pdf

Cern: Open Data. Recuperado el 3 de octubre de 2016 desde <http://opendata.cern.ch/>

Coalición Proacceso Retrieved (October 23, 2016) Recuperado desde <http://www.proacceso.org/>

Cuadrado, J. J., & Monzón, M.D. (2016). Analysis of the impact of file formats for open data analytics efficiency: A case study with R. GSTF Journal on Computing (JoC), 5(1), 40-44. doi:http://dx.doi.org/10.5176/2251-3043_4.4.351

Datos abiertos GenCat. Recuperado el 3 de octubre de 2016 desde <http://dadesobertes.gencat.cat/es>

Datos.gob.es: Catálogo de datos. Recuperado el 3 de octubre de 2016 desde <http://datos.gob.es/catalogo>

Dewailly, L. (2015). Building a RESTful Web Wervice with Spring: A hands-on guide to building an enterprise-grade, scalable RESTful web service using the Spring Framework. Birmingham: Packt Publishing.

Enel Open Data. Recuperado el 3 de octubre de 2016 desde <http://data.enel.com/>

García, S. Á., Barrio, M. G., & Fernández, M. R. (2014). The collaborative construction of open databases as tools for citizen empowerment. Revista Latina De Comunicación Social, (69), 661-683. doi:<http://dx.doi.org/10.4185/RLCS-2014-1029en>

Hamza Ed-Douibi. Modeling Web APIs: your best choices. Recuperado el 29 de noviembre de 2016 desde <http://modeling-languages.com/modeling-web-api-comparing/>

Havens, A. Beginners guide to creating a REST API. Recuperado el 31 de diciembre de 2016, desde <http://www.andrewhavens.com/posts/20/beginners-guide-to-creating-a-rest-api/>

Hayashi, T., & Ohsawa, Y. (2016). Comparison between Utility Expectation of Public and Private Data in the Market of Data. Procedia Computer Science, 96, 1267-1274. doi:10.1016/j.procs.2016.08.171

JSON. En Wikipedia. Recuperado el 23 de octubre de 2016 desde <https://es.wikipedia.org/wiki/JSON>

Karzyński, P. B. Building beautiful RESTful APIs using Flask, Swagger UI and Flask-RESTPlus. Recuperado el 31 de diciembre de 2016, desde <http://michal.karzynski.pl/blog/2016/06/19/building-beautiful-restful-apis-using-flask-swagger-ui-flask-restplus/>

Kitchin, R. (2014). The data revolution: Big data, open data, data infrastructures & their consequences. Los Angeles, CA: SAGE Publications.

Licenses. En Open Data Commons. Recuperado el 29 de octubre de 2016 desde <http://opendatacommons.org/licenses/>

Linked Data. En W3.org. Recuperado el 1 de noviembre de 2016 desde <https://www.w3.org/DesignIssues/LinkedData.html>

Market Place: explore. En Mashape. Recuperado el 3 de diciembre de 2016 desde <https://market.mashape.com/explore>

Mazaheri, M. (2015) Tools that will help you develop a RESTful API. Recuperado el 31 de diciembre de 2016 desde <https://blog.paw.cloud/posts/tools-for-restful-api/>

Murray-Rust, P. (2008, 03). Open Data in Science. Serials Review,34(1), 52-64. doi:10.1016/j.serrev.2008.01.001

Nielsen M (2011) Reinventing discovery: the new era of networked science. Princeton University Press

Open data en el sector privado, 2010. Recuperado el 10 de octubre de 2016 desde <http://www.digitalassetsdeployment.com/noticias/open-data-el-sector-privado>

Open Data en Gobierno de Navarra. Recuperado el 1 de noviembre de 2016 desde <http://www.gobiernoabierto.navarra.es/es/open-data/datos/formatos>

Open Data Handbook: value stories. Recuperado el 3 de noviembre de 2016 desde <http://opendatahandbook.org/value-stories/en/>

OpenDataBCN. Recuperado el 3 de octubre de 2016 desde <http://opendata.bcn.cat/opendata/es>

Portal de datos abiertos de la Unión Europea. Recuperado el 3 de octubre de 2016 desde <https://data.europa.eu/euodp/es/data>

ProgrammableWeb. (2013). ProgrammableWeb: What Programming Language is Most Popular with APIs?. Recuperado el 24 de noviembre desde <https://www.programmableweb.com/news/what-programming-language-most-popular-apis/2013/06/03>

Richardson, L., & Amundsen, M. (2013). RESTful Web APIs. Sebastopol, CA: O'Reilly.

Rinciog, O., & Posea, V. (2016). Lodro: Using cultural Romanian Open Data to build new learning applications. Paper presented at the , 1 267-274. doi:<http://dx.doi.org/10.12753/2066-026X-16-038>

Rodríguez, C., Baez, M., Daniel, F., Casati, F., Trabucco, J. C., Canali, L., & Percannella, G. (2016). REST APIs: A Large-Scale Analysis of Compliance with Principles and Best Practices. Lecture Notes in Computer Science Web Engineering, 21-39. doi:10.1007/978-3-319-38791-8_2

Sandoval, K. (2015). Top Specification Formats for REST APIs | Nordic APIs |. Recuperado el 29 de diciembre de 2016 desde <http://nordicapis.com/top-specification-formats-for-rest-apis/>

Say, M. En CIO. Recuperado el 3 de noviembre de 2016 desde <http://www.cio.co.uk/it-strategy/can-open-data-work-for-private-sector-3494669/>

Sede-e: catálogo de datos. Recuperado el 3 de octubre de 2016 desde <https://sede.minetur.gob.es/es-ES/datosabiertos/catalogo>

Shapefile. En Wikipedia. Recuperado el 23 de octubre de 2016 desde <https://es.wikipedia.org/wiki/Shapefile>

Silberschatz, A., Sudarshan, S., Korth, H. F., & Pérez, F. S. (2002). Fundamentos de bases de datos. Madrid: McGraw-Hill.

Skelton, C. (2011, Jul 20). B.C. launches open-data site; nearly 2,500 sets of government info available for download. The Vancouver Sun. Recuperado desde <http://search.proquest.com/docview/878703028?accountid=14777>

Stephens, O. (2011). Mashups and open data in libraries. Serials, 24(3), 245-250. Retrieved from <http://search.proquest.com/docview/907239751?accountid=14777>

The Next Frontier for Open Data: An Open Private Sector. Recuperado el 01 de noviembre de 2016 desde <https://blogs.worldbank.org/voices/next-frontier-open-data-open-private-sector>

The open university embraces the linked open data movement. (2010, Nov 11). Targeted News Service Recuperado desde <http://search.proquest.com/docview/818569408?accountid=14777>

Valencia: Datos abiertos. Recuperado el 3 de octubre de 2016 desde <http://www.valencia.es/ayuntamiento/datosabiertos.nsf/>

Valores separados por comas. En Wikipedia. Recuperado el 23 de octubre de 2016 desde <https://es.wikipedia.org/wiki/CSV>

Van Noorden, R. (2014). Confusion over open-data rules. Nature, 515(7528), 478. Recuperado desde <http://search.proquest.com/docview/1632527244?accountid=14777>

What is Open Data. En Open Data Handbook. Recuperado el 23 de octubre de 2016 desde <http://opendatahandbook.org/guide/es/what-is-open-data/>

XI Jornadas internacionales gvSIG. CNIG. Recuperado en 30 de diciembre del 2016 desde http://www.gvsig.com/es/eventos/jornadas-gvsig/11as-jornadas-gvsig/comunicaciones/-/asset_publisher/Xgz02FK0m1bJ/content/open-data-legal-and-technical-aspects

Young, B. (2016). Virtual Panel: Document and Description Formats for Web APIs. Recuperado el 28 de diciembre de 2016, desde <https://www.infoq.com/articles/document-description-formats-web-apis>

5 empresas con proyectos basados en Open Data. Recuperado el 31 de octubre de 2016 desde <http://openbigdatamanagement.com/5-empresas-con-proyectos-basados-en-open-data/>