GlobalLogic

Comparación de API y SOA

Maximiliano García Santanna
Sr. Developer Engineering

Buenos Aires, Argentina

Un documento que clarifica el concepto de API como solución tecnológica actual y cómo se compara a SOA.

www.GlobalLogic.com.ar

Maximiliano García Santanna
Sr. Developer Engineering

Índice

Maximiliano García Santanna
Sr. Developer Engineering

¿Qué es una API?

Se define API (del inglés: *Application Programming Interface*), como el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizada por otro software como una capa de abstracción.

Una API representa la capacidad de comunicación entre componentes de software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general. Las API asimismo son abstractas: el software que proporciona la funcionalidad de una cierta API, es la implementación de esa API.

Esto no es nada nuevo, así que la pregunta que surge es ¿A que se le llama API ahora?

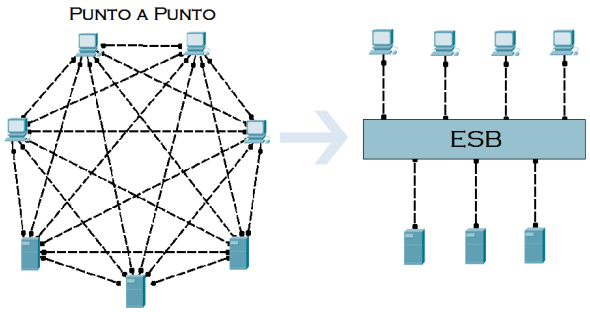
Maximiliano García Santanna
Sr. Developer Engineering

¿A qué se le llama API ahora?

Actualmente API es una evolución de SOA, si bien la implementación desde el punto de vista tecnológico no tienen grandes diferencias, es necesario entender el contexto en el que nace cada uno para poder entender las diferencias. Ambos fueron concebidos con una orientación y conceptos muy distintos.

Maximiliano García Santanna
Sr. Developer Engineering

Surgimiento de SOA



SOA surgió como solución en un contexto de gran cantidad de conexiones punto a punto. Se ideó enfocado en los proveedores de servicio, con el fin de reducir costos y el esfuerzo de agregar un nuevo consumidor y reutilizar funciones comunes, encapsulando capacidades en servicios. Cada servicio posee una interfaz y representa una capacidad, si se quiere acceder a esa capacidad, se debe acceder mediante ese servicio. De esta manera se pueden agregar consumidores sin modificar la interfaz, ocultando a los consumidores la implementación, que podría cambiar. Se definieron "buenos servicios" desde un enfoque de proveedor: "*Esto es lo que tengo para ofrecer*".

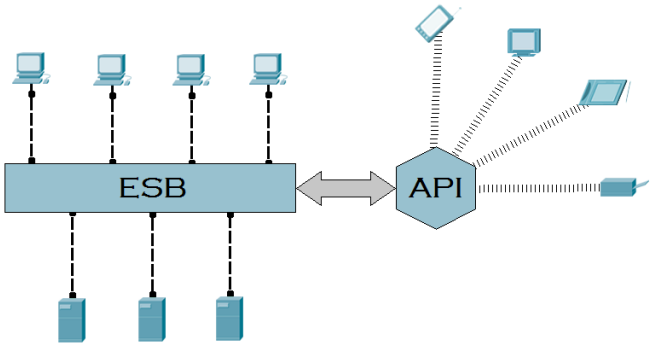
Los servicios se diseñan con una granularidad media, para que con uno se puedan hacer bastantes cosas, pero no demasiadas para que no sea abrumador, y tampoco muy pocas, para que no haya que consumir distintos servicios para una única necesidad. En este enfoque de proveedor, se concentraron los esfuerzos en los desafíos de conectividad, para mapear los proveedores con los consumidores, brindando estabilidad y seguridad. Pensado para realizar dos o tres proyectos al año, que reutilizarían las funciones encapsuladas y que ocasionalmente se debería modificar algo en el back-end de los servicios.

Maximiliano García Santanna
Sr. Developer Engineering

Surgimiento de API

API surge con una realidad mega conectada y cambiante. Cada vez más, los consumidores se relacionan con los proveedores desde dispositivos o contextos no tradicionales. El mundo Mobile y Social está creciendo exponencialmente y las aplicaciones mobile por ejemplo, reciben varias actualizaciones por mes, mucho más rápido que cualquier cosa que se hubiese tenido antes. Estos medios de interacción, son llamados Sistemas de Compromiso, son dispositivos y contextos que relacionan y generan compromiso a los clientes con el negocio. Si tratamos de satisfacer esta creciente demanda de interacción con un enfoque SOA, tendríamos que cambiar constantemente los servicios para añadir las nuevas capacidades e información que los sistemas de compromiso quieren y es muy difícil, no porque técnicamente no se pueda, sino que aumenta la probabilidad de introducir errores al sistema por la cantidad y la velocidad de los cambios. No se podrían correr todos los test de regresión a tiempo y no se podría asegurar la integridad del código.

Maximiliano García Santanna
Sr. Developer Engineering



Estos servicios son el core del Negocio, los que lo mantienen funcionando, así que no se pueden correr riesgos en cuanto a estabilidad y disponibilidad de los mismos. Es necesario encapsular esos sistemas, aun más de lo que SOA lo hace y permitir a los sistemas de compromiso que consuman la información en la forma que quieran, lo más rápido posible, sin hacer modificaciones en el back-end.

Así es como nace API, no como un reemplazo a SOA, sino como un complemento al mismo, para cubrir una demanda más ágil y particular.

PLATAFORMA API

El sistema de gestión de API consta de tres componentes que se describen en las siguientes secciones:

API Gateway

A continuación se detallan las responsabilidades de este componente.

- La responsabilidad principal es la de realizar la interconexión entre los servicios y los consumidores, a través de las API publicadas en él.

- Ruteo: enrutamiento de mensajes a diferentes destinos en base a un conjunto de condiciones y el contexto del mensaje.

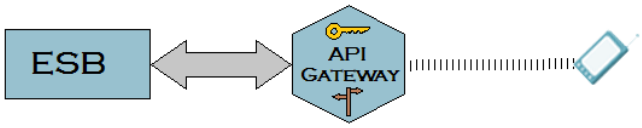
- Soporte de múltiples formatos: es responsable de la transformación de datos de un formato a otro (JSON y XML principalmente).

- Soporte para múltiples protocolos: debe soportar varios protocolos, tanto para la publicación de las API como para la comunicación con los servicios internos.

- Monitoreo: Monitorización del tráfico de entrada y salida.

- Políticas de seguridad: Otorga a las API autorización, autenticación y cifrado de los mensajes utilizando los estándares de tecnologías conocidas (oAuth).

API Manager



A continuación se detallan las responsabilidades de este componente:

- La responsabilidad principal es ofrecerle a los proveedores de API la capacidad de configurar y publicar sus API en el componente de API Gateway.

- Publicación: publicar las API en API Gateway, definiendo su punto de entrada (URL por ejemplo) para permitir el acceso al mismo.

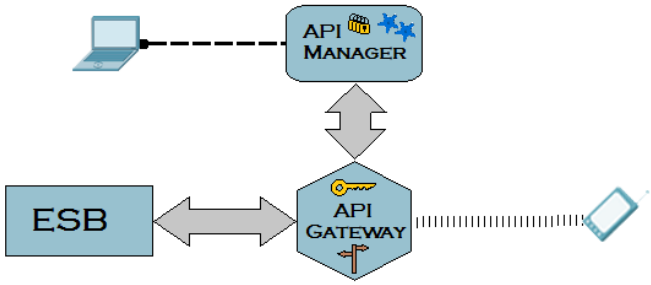
- Edición: Herramientas para el diseño de la interfaz de la API.

- Control de tráfico: Monitorización del uso de las API, y sistema de configuración de control según parámetros del consumo.

- Gestor de políticas de seguridad: Sistema de configuración de seguridad de las API.

- Gestor de ciclo de vida: versionado y deprecado de las API.

API Portal



A continuación se detallan las responsabilidades de este componente:

- La responsabilidad de este componente, es la de recopilar toda la información necesaria para los consumidores sobre las API publicadas en el API Gateway (Documentación, comunidad, análisis).

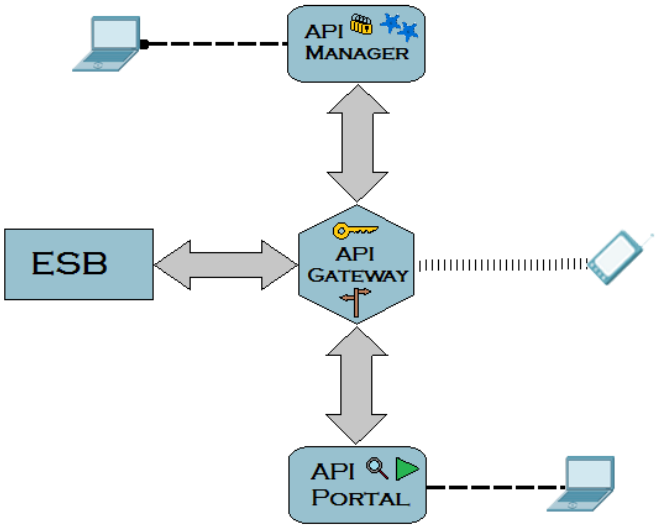
- Prueba: Sistema para la prueba de las API.

- Navegador: buscador de las API registradas con filtros, como estado, versión, nombre, etc.

- Análisis de uso: Monitoreo y análisis del uso de las API, status de los resultados, tiempo promedio de respuesta, fallos, etc.

Maximiliano García Santanna
Sr. Developer Engineering

Usos y Características



Para las API , la prioridad es lo que quiere el consumidor. El primer caso de uso que se suele plantear a la hora de crear una API, es ¿qué quiere el desarrollador Mobile de la empresa?

De esta manera, se publica la API y se hace disponible en el portal de auto-servicio (API Portal) en el que el desarrollador puede ingresar, encontrar la API, entenderla fácilmente y registrarse para usarla.

Cuando se crea la API, se crean ejemplos de entrada y salida para que pueda ser utilizado en el desarrollo que estén realizando y así estar listos para consumirla en muy poco tiempo. Es un enfoque de consumo rápido.

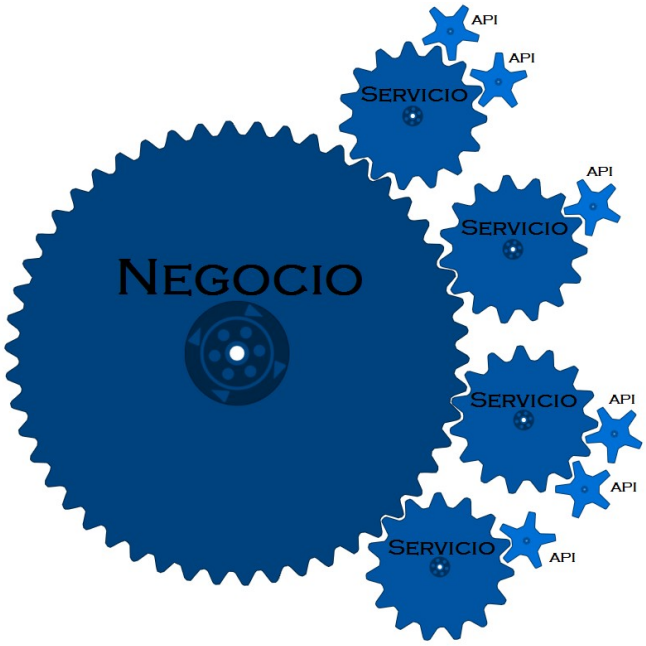
Como el primer caso de uso suele ser el desarrollador mobile, se tienen que tener en cuenta ciertos aspectos. La API debe ser algo que se pueda utilizar en el dispositivo en el que va a correr, un celular o tablet, es decir, algo con batería, así que no puede ser algo que consuma toda la batería, porque esto haría que la aplicación fracase. Tiene que ser ligera, de granularidad muy fina, para no tener que buscar en un conjunto grande de datos puntualmente el que se necesita, sino obtener la respuesta muy rápidamente y sin esfuerzo.

Hay más foco en qué pasa luego del despliegue, que antes. No se dedica demasiado tiempo de gobierno en tiempo de diseño o en los aspectos de reutilización, la idea detrás de las API, es que no importa si se crea una API por consumidor, porque son pequeñas, livianas y rápidas de crear.

El objetivo de API siempre es centrado en los consumidores, por lo que por ejemplo, se puede implementar para integrar nuevos partners. La mayoría se relacionan con el diseño pre-existente de SOA, pero un enfoque para sumar a la minoría restante, es hacerlo mediante API.

Maximiliano García Santanna
Sr. Developer Engineering

Diferencias entre API y SOA



Como se menciona anteriormente, entre API y SOA hay similitudes en cuanto a la implementación tecnológica.

Maximiliano García Santanna
Sr. Developer Engineering

En el mundo SOA los Web Services son muy comunes y también existen tecnologías que soportan REST, por otro lado API Management se implementa principalmente con REST y también hay soporte para Web Services. Tanto en SOA como en API se habla de consumidores y proveedores y hay algún nivel de transformación. Entonces ¿Cuál es la diferencia?

La principal diferencia es el propósito que hay debajo de la tecnología.

Las API son servicios, pero no todos los servicios son APIs. En SOA se encapsulan las capacidades dentro del sistema y de esta manera se expone todo lo que se puede proveer. Con API, se intenta hacer lo opuesto, está enfocado en exponer lo que quiere el consumidor.

Si ya se tiene un gobierno SOA, es un punto de partida perfecto para crear APIs, porque ya se cuenta con los servicios que exponen lo que se puede proveer, ahora sólo resta pensar que quieren tus consumidores y crear APIs que transformen lo que se puede proveer en lo que se quiere consumir. Esto no quiere decir explotar 100 servicios en 500 APIs, porque eso seguiría siendo una perspectiva de proveedor y no de consumidor.

API descarta toda la información que no necesita de un servicio y ese es el nivel de ruteo y transformación que se busca con API, no las transformaciones complicadas que hay en ESB.

Maximiliano García Santanna
Sr. Developer Engineering

En SOA, se invierte gran parte del tiempo de gobierno en la etapa de diseño, mientras que en API el gobierno está en administrar lo que pasa luego de que se despliega la API.

En SOA se aplica un proceso robusto de gobierno, con metodologías para identificar la granularidad correcta de los servicios y énfasis en la definición de las interfaces. Por otro lado, API se enfoca en crear y liberar rápidamente las API y enfoca su tiempo de gobierno en el ciclo de vida de la misma.

Ventajas de API

Como se dijo previamente, se podría encarar esta masificación de sistemas de compromiso con SOA, pero no se contaría con las siguientes ventajas que se desprenden del uso de API.

- Competitividad: La primera y considero más importante ventaja es la competitividad. Por ejemplo, existen muchas aplicaciones que comparan productos, y si se quiere que el negocio forme parte de esa comparativa, se tienen que crear APIs que le permitan a los desarrolladores, obtener los datos a comparar, rápida y fácilmente, para poder integrar los datos de negocio, sino se pierde una potencial oportunidad de ganar clientes.

- Ampliar visibilidad: Otra de las ventajas, es la de ampliar la visibilidad de la empresa y por ende un incremento en los clientes, apuntando a negocios que tengan servicios que se complementen o se puedan relacionar con los servicios que ofrece nuestro negocio.

Un claro ejemplo me parece MercadoPago. Éste puede ponerse en contacto con negocios en los que haya transacciones de dinero entre dos partes, y ofrecerles su API.

De esta manera, MercadoPago amplía el alcance ganando más clientes y la empresa que lo ofrece como opción, genera una mejor experiencia de pago para sus clientes, haciendo su oferta más atractiva que la de un competidor que no la ofrezca. Es un *win-win.*

- Tecnologías y mercados emergentes: estar preparado para las distintas tecnologías y mercados emergentes es fundamental para el crecimiento de cualquier negocio. La tendencia de *Internet of Things* (Internet de las Cosas), que al igual que Mobile, suelen ser artefactos con batería o procesamiento limitados, también se ven beneficiados del uso de API. Un ejemplo podrían ser los equipos de aire acondicionado, que se pueden controlar desde internet.

- Desacoplamiento con la Interfaz: Mediante API se puede separar la interfaz del modelo de datos y despreocuparse de si se lo está mostrando en un explorador web, un teléfono celular o cualquier otro dispositivo que exista (o vaya a existir), como *Google Glass.*

Reflexión Final

SOA estableció los cimientos para el nuevo enfoque de API, fomentando estándares abiertos y una cultura orientada a servicios, preparando a la organización para pensar y colaborar en los propósitos de negocio que van más allá del silo de la organización.

La industria actualmente no está descartando SOA. SOA no es una herramienta, es un marco de trabajo y un modelo de arquitectura. SOA en su estado más puro, sigue vigente, lo que cambió es cómo se consumen los servicios en la actualidad debido a nuevas necesidades en una era de productos orientados al consumidor, para los cuales el enfoque SOA no es totalmente apropiado.

API toma las lecciones aprendidas de SOA y evoluciona el enfoque de orientación a servicios para la realidad de hoy en día.

Sobre GlobalLogic

GlobalLogic es una empresa líder en el desarrollo completo del ciclo de vida de los productos de software. Combinando experiencia en diversas industrias y expertise en nuevas tecnologías, ayudamos a nuestros clientes a conectar ideas innovadoras con resultados de negocio. A partir del conocimiento obtenido en la creación de productos de software innovadores con tecnologías de vanguardia, proveemos servicios de consultoría y desarrollo en areas como Mobile, Cloud Computing, SaaS, UX Design, Rich Internet Applications, Social Media, SOA&BPM, entre otros. A través de una estrecha colaboración con nuestros clientes, los ayudamos a responder a las exigencias del time to market y a lograr costos competitivos en cada fase del ciclo de vida del desarrollo.

GlobalLogic

Contacto

Daniela Castelli
+54.11.5533.8300 x 1016
daniela.castelli@globallogic.com