



# Arquitectura de integración ágil

*Ejecutables de integración livianos, alineados con microservicios y basados en contenedores*

---

## Contenido

- 1 Resumen ejecutivo
  - 2 La integración ha cambiado
  - 2 El cambio hasta el momento: SOA, ESB y API
  - 3 Una arquitectura de integración ágil
  - 3 Aspecto 1: Implementación detallada de la integración
  - 4 Aspecto 2: Propiedad descentralizada de la integración
  - 5 Aspecto 3: infraestructuras de integración nativas de la nube
  - 5 ¿De qué manera el ejecutable de integración moderna ha cambiado para adaptarse a la arquitectura de integración ágil?
  - 6 Arquitectura de integración ágil para la plataforma de integración
  - 11 La plataforma IBM® Cloud Integration
- 

Adopte la transformación digital con una integración ágil centrada en un enfoque también ágil, con la que podrá moverse rápidamente y satisfacer las demandas de nubes múltiples, descentralización y microservicios.

## Resumen ejecutivo

Las organizaciones que buscan la transformación digital deben adoptar nuevas formas de usar e implementar las tecnologías de integración, para moverse rápidamente y de acuerdo a los objetivos de nubes múltiples, descentralización y microservicios. La capa de integración de aplicaciones debe transformarse a fin de que las organizaciones puedan crear adecuadamente nuevas experiencias del cliente, en lugar de forzar modelos de arquitectura y desarrollo que no fomentan el aumento de la productividad de la organización.

Muchas organizaciones han empezado con la adopción de técnicas de aplicación ágiles, como la arquitectura de microservicios, y están actualmente comenzando a disfrutar de los beneficios que les aparejó dicho cambio. Este tipo de enfoque complementa y acelera la estrategia de API de las empresas, que también deberían usar este enfoque para modernizar su infraestructura ESB existente y así desarrollar formas más eficaces de gestionar y operar sus servicios de integración en la nube privada o pública.

Este documento técnico se basa en un [libro](#) que analiza los méritos de lo que denominamos **arquitectura de integración ágil**, un enfoque basado en contenedores, alineado con microservicios y descentralizado para integrar soluciones que satisfacen las exigencias de agilidad, escalabilidad y resiliencia que demanda la transformación digital.



## La integración ha cambiado

IDC calcula que el gasto en iniciativas relacionadas con la transformación digital representará una oportunidad de mercado de USD 20 billones en los próximos cinco años<sup>1</sup>.

¿Qué hay detrás de esta impactante inversión? La necesidad siempre presente y en aumento de crear nuevas experiencias del cliente mediante experiencias conectadas en una red de aplicaciones que aprovechan los datos de todo tipo.

Y no es una tarea fácil, ya que combinar procesos y fuentes de datos en el momento correcto y en el contexto apropiado es algo difícil, en especial si se tiene en cuenta la adopción competitiva de aplicaciones empresariales SaaS. Se deben integrar nuevas fuentes de datos en los procesos de negocio para crear una diferencia competitiva.

---

*“Para impulsar nuevas experiencias de cliente, las organizaciones deben tener acceso a un conjunto de aplicaciones, procesos y fuentes de información en constante crecimiento, lo que amplía en forma significativa la necesidad de la empresa de tener capacidades de integración e invertir en ellas”.*

### El valor de la integración de las aplicaciones para la transformación digital

En el momento de plantearse cómo crear nuevas experiencias de cliente y centrarse en el acceso a los datos y su disponibilidad para los servicios y las API que alimentan estas iniciativas, hay muchos beneficios importantes que la integración de aplicaciones pone a disposición.

- Direccionamiento eficaz de la disparidad: acceso a los datos desde cualquier sistema y en cualquier formato, y crear homogeneidad a partir de esto, sin importar la diversidad del entorno de múltiples nubes.
- Conocimiento experto de los puntos finales: la integración moderna incluye inteligencia relacionada con los protocolos complejos y formatos de datos, pero también incorpora inteligencia de los objetos reales, la empresa y las funciones de los sistemas finales.

- Innovación mediante datos: las aplicaciones basan gran parte de su innovación en la oportunidad de combinar datos que superen sus límites y crear significativamente a partir de allí, una característica especialmente visible en la arquitectura de microservicios.
- Artefactos de nivel empresarial: los flujos de integración heredan un número enorme de valor del ejecutable, en el que están incluidas características de nivel empresarial para recuperación de errores, tolerancia a fallos, captura de registros, análisis de rendimiento y mucho más.

El panorama de integración se está modificando para adaptarse a las exigencias informáticas de las empresas y el mercado, ¿pero cómo hicimos para transformarnos de SOA y ESB en una arquitectura moderna, de integración ágil y basada en contenedores?

### El cambio hasta el momento: SOA, ESB y API

Antes de poder avanzar hacia el futuro de la integración ágil, necesitamos comprender qué hubo antes. Los patrones de SOA (arquitectura orientada a servicios) surgieron a comienzos del milenio, y en un principio, la amplia aceptación de los estándares en los que se basaban hacían pensar en un futuro brillante, donde todos los sistemas podían descubrirse y comunicarse con otros sistemas mediante patrones de exposición síncrona.

Si avanzamos un poquito en el tiempo, estamos en el medio del movimiento de la tecnología ESB (bus de servicio empresarial), que en teoría iba a proporcionar conectividad a sistemas de backend, que provenían del patrón anterior de “centro y periferia”. Si bien muchas empresas implementaron con éxito el patrón ESB, el término no es exactamente apreciado en el entorno nativo de la nube, ya que se considera pesado y sin agilidad. ¿Cómo hacemos, entonces, para ir de un extremo al otro?

### La realidad se reduce a unos pocos factores, muchas veces relacionados entre sí:

- SOA era más compleja que la implementación de un ESB, en especial en lo concerniente a quién debía financiar un programa para toda la empresa.
- Los patrones de ESB formaban una sola infraestructura para toda la empresa, con decenas o cientos de integraciones instaladas en un clúster de servidores en producción. Si bien el patrón ESB no exige una centralización importante, las topologías resultantes generalmente son así.

<sup>1</sup>IDC MaturityScape Benchmark: Digital Transformation Worldwide, 2017, Shawn Fitzgerald. Golluscio.

- Con frecuencia, los patrones ESB centralizados no entregaban los ahorros importantes que esperaban las empresas, ya que las interfaces no se podían usar de un proyecto a otro.
- Las iniciativas entre empresas, como los patrones ESB, tenían problemas para encontrar financiación, y muchas veces esta se aplicaba únicamente a los servicios que podrían reutilizarse lo suficiente como para cubrir los costos de su creación

---

*Los patrones ESB han tenido problemas para asegurar la financiación continua de iniciativas a nivel de toda la empresa, ya que no se aplican específicamente en el contexto de una iniciativa empresarial.*

---

El resultado fue que la creación de servicios por parte de este equipo de especialistas SOA se convirtió en un cuello de botella para los proyectos, en lugar de ser el facilitador que debía ser. Esto hizo que el patrón ESB centralizado tuviera una mala reputación también.

La arquitectura orientada a los servicios aplicada a los patrones ESB es una iniciativa de toda la empresa para crear servicios y API reutilizables y disponibles de manera síncrona, de forma que se puedan crear nuevas aplicaciones más rápidamente incorporando datos de otros sistemas

La arquitectura de microservicios, por otro lado, es una opción respecto de cómo podría escribir una aplicación individual de manera que sea más ágil, escalable y resiliente.

## **Una arquitectura de integración ágil**

¿Por qué se han popularizado tanto los conceptos relacionados con los microservicios en el ámbito de las aplicaciones?

Representan un enfoque alternativo a la estructuración de las aplicaciones. En lugar de que la aplicación sea un enorme silo de código que se ejecuta en el mismo servidor, la aplicación se diseñó como una colección de componentes más pequeños de ejecución totalmente independiente.

La arquitectura de microservicios permite tres beneficios cruciales:

1. Mayor **agilidad**: los microservicios son tan pequeños que se los comprende totalmente en aislamiento y se los cambia de forma independiente.

2. **Escalabilidad** elástica: el uso de los recursos se integra totalmente en el modelo empresarial.

3. **Resiliencia** discreta: con un desacople apropiado, los cambios hechos en un microservicio no afectan a los demás en el ejecutable.

Si se tienen en cuenta estos beneficios, ¿cómo sería una integración rediseñada, que se implementa normalmente en silos centralizados, con una perspectiva nueva y basada en una arquitectura de microservicios? A esto llamamos **“arquitectura de integración ágil.”**

---

*La arquitectura de integración ágil se define como “una arquitectura basada en contenedores, descentralizada y alineada con microservicios para brindar soluciones de integración”.*

---

La arquitectura de integración ágil tiene tres aspectos relacionados, pero independientes:

### **Aspecto 1: Implementación detallada de la integración.**

¿Qué ganaríamos si dividimos las integraciones del silo de ESB en ejecutables independientes?

### **Aspecto 2: Propiedad descentralizada de la integración.**

¿Cómo debemos modificar la estructura organizativa para aprovechar mejor un enfoque más detallado?

### **Aspecto 3: Infraestructuras de integración nativas de nube.**

¿Qué beneficios adicionales podemos obtener con una integración basada totalmente en un enfoque nativo de nube?

## **Aspecto 1: Implementación detallada de la integración**

La implementación descentralizada del centro de integración o los patrones ESB donde todas las implementaciones se despliegan en un único par de servidores de integración de alta disponibilidad (HA) ha demostrado ser responsable de la creación de cuellos de botella para los proyectos.

Cualquier implementación en los servidores compartidos presenta el riesgo de desestabilizar las interfaces críticas existentes. Ningún proyecto individual puede actualizar la versión del middleware de integración para acceder a nuevas funciones.

Podríamos desglosar el componente ESB en el nivel de toda la empresa en fragmentos más pequeños, gestionables y dedicados. Quizás en algunos casos podemos incluso llegar hasta un ejecutable por cada interfaz que exponemos. Estos patrones de “implementación detallada de la integración” proporcionan contenedores especializados del tamaño adecuado, que ofrecen mayor agilidad, escalabilidad y resiliencia, y que se ven muy distintos de los patrones ESB centralizados del pasado. En la Figura 1 se muestra de manera simple la diferencia entre un ESB centralizado y una implementación detallada de la integración.

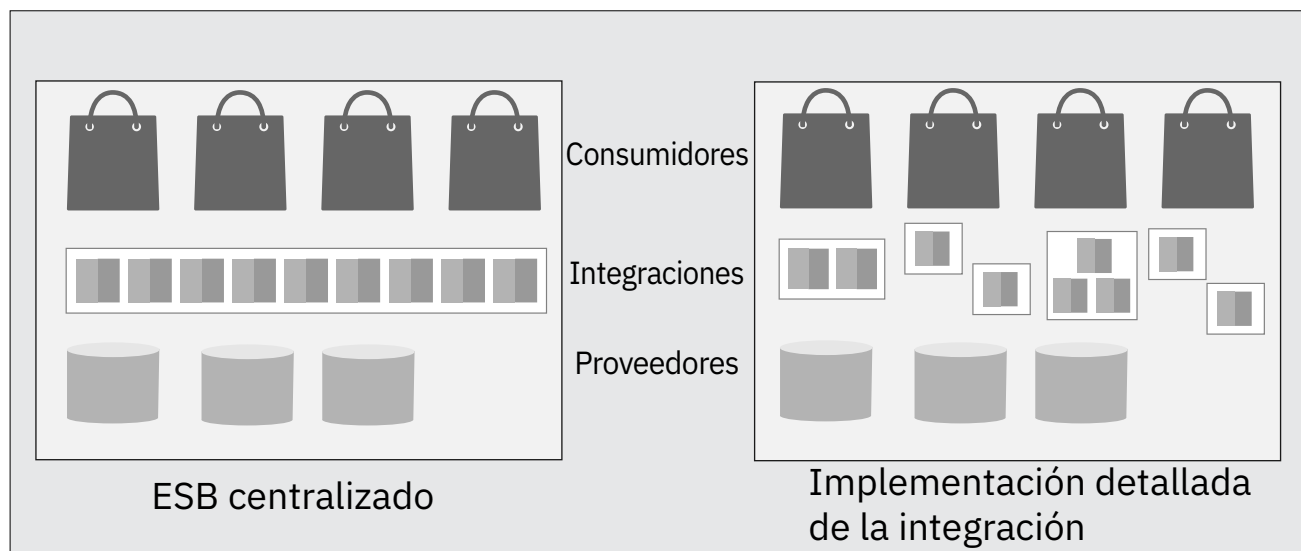


Figura 1: Comparación sencilla de un ESB centralizado con una implementación detallada de la integración

La implementación detallada de la integración aprovecha los beneficios de la arquitectura de microservicios. Veamos cuáles son los beneficios de los microservicios en vista de la implementación detallada de la integración:

- **Agilidad:** diferentes equipos pueden trabajar independientemente en las integraciones, sin derivar a un grupo centralizado o una infraestructura que pueda producir rápidamente un cuello de botella. Los flujos individuales de integración se pueden modificar, reconstruir e implementar independientemente de los demás flujos, lo que permite cambios más seguros mejora de la velocidad de la producción.
- **Escalabilidad:** los flujos individuales pueden escalarse independientemente, lo que le permite aprovechar el escalado eficiente y elástico de las infraestructuras de la nube.
- **Resiliencia:** los flujos de integración aislados que se implementan en contenedores independientes no pueden influir entre sí robándose recursos compartidos, como la memoria, las conexiones o el CPU.

Cuando se piensa en agilidad, escalabilidad y resiliencia, es importante tener presente que no pueden obtener estos beneficios de la integración detallada sin la integración descentralizada.

[Obtenga mucha más información](#) acerca de la integración detallada en nuestro libro Agile Infrastructure Architecture, disponible actualmente para [descargar](#).

## Aspecto 2: Propiedad descentralizada de la integración

Un desafío importante al que se enfrentaba la arquitectura orientada a servicios era que tendía a forzar la creación de equipos de integración central, y la infraestructura para crear la capa de servicios.

Esto creaba una fricción continua en el ritmo al que se podían ejecutar los proyectos, porque siempre tenían al equipo de integración central como dependencia. Este conocía bien la tecnología de integración, pero muchas veces no comprendían las aplicaciones que integraban, por lo que la traducción de los requisitos podía ser lenta y producir errores.

Muchas organizaciones habrían preferido que los equipos de aplicaciones estuvieran a cargo de la creación de sus propios servicios, pero la tecnología y la infraestructura existente en ese momento no lo permitieron.

El paso a una implementación detallada de la integración plantea la posibilidad de que la propiedad de la creación y el mantenimiento de las integraciones pueda distribuirse. No es tan descabellado que los equipos de aplicaciones empresariales se ocupen del trabajo de integración, optimizando la implementación de nuevas capacidades.

¿Se despertó su curiosidad respecto de la implementación de una integración detallada? Encuentre todas las respuestas a sus preguntas en el libro [Agile Infrastructure Architecture \(Arquitectura de infraestructura ágil\)](#), ya disponible.

### **Aspecto 3: Infraestructuras de integración de nube nativas**

Los tiempos de integración han cambiado drásticamente en estos últimos años. Tal es así que estas ejecuciones ligeras se pueden utilizar de formas verdaderamente nativas de nube. Es decir, tienen capacidad para rechazar la carga de muchos de sus anteriores mecanismos propietarios para la gestión de clústeres, el escalado, la disponibilidad y de la plataforma de la nube en la que se ejecutan.

Todo esto implica mucho más que ejecutarlas en un ambiente en contenedor, ya que deben funcionar como “ganado” y no como “mascotas”, aprovechando de mejor forma las capacidades de organización, como servicios Kubernetes y muchos otros contextos comunes y estándares en la nube.

---

*La adopción de un “enfoque de ganado” influye en cómo los equipos de DevOps interactuarán con el entorno y la solución global, y creará mayores eficiencias a medida que más soluciones se trasladen a las arquitecturas ligeras.*

---

### **¿De qué manera el ejecutable de integración moderna ha cambiado para adaptarse a la arquitectura de integración ágil?**

Indudablemente, la arquitectura de integración ágil requiere que la topología de integración se implemente de manera muy diferente, y un aspecto clave de esto es un ejecutable de integración moderno que pueda ejecutarse en un entorno basado en contenedor, adaptado correctamente a las técnicas nativas de implementación de nube. Los tiempos de ejecución de integración modernos apenas si guardan relación con sus versiones anteriores. Veamos algunas de estas diferencias:

- **Ejecutables ligeros y rápidos:** funcionan en contenedores como Docker, y son lo suficientemente ligeros para comenzar y detenerse en segundos, y se pueden administrar fácilmente mediante entornos de coordinación, como los Kubernetes.
- **Sin dependencias:** ya no necesitan de bases de datos o colas de mensajes, si bien son expertos en conectarse a estas si es necesario.
- **Instalación basada en sistemas de archivos:** Pueden instalarse colocando los binarios en un sistema de archivos y haciéndolos arrancar. Esto es perfecto para los sistemas de archivos en capas de las imágenes de Docker.
- **Soporte para herramientas para DevOps:** el ejecutable debe estar listo para la integración y la implementación continuas. Instalación, compilación, implementación y configuración basados en archivos de propiedades y scripts, a fin de permitir prácticas de “infraestructura como código”. Para acelerar la inclusión en los canales de DevOps, se deben proporcionar plantillas de scripts para herramientas estándares de compilación e implementación.
- **API primero:** El protocolo de comunicación principal debe ser con API RESTful. La exposición de las integraciones como API RESTful debe ser trivial y estar basada en convenciones comunes, como la especificación API Open. Además, llamar las API RESTful también debe ser trivial, incluido el descubrimiento mediante archivos de definición.
- **Conectividad digital:** junto con la potente conectividad empresarial que los ejecutables de integración siempre han proporcionado, también deben conectarse a recursos modernos. Por ejemplo, bases de datos NoSQL (MongoDb, Cloudant, entre otras) y servicios de mensajería como Kafka. Es más, deben tener acceso a un extenso catálogo de conectores inteligentes de aplicación para SaaS (software como servicio), como Salesforce.

- **Entrega continua:** la prestación continua se logra mediante interfaces de línea de comandos y plantillas de scripts que se combinan con las herramientas estándares de canal de DevOps. Esto reduce incluso más el conocimiento necesario para implementar interfaces y aumentar el ritmo de entrega.
- **Herramientas mejoradas:** El uso de herramientas mejoradas para la integración significa que pueden construirse más interfaces solamente con configuración, muchas veces realizado por personas sin conocimientos de integración. Con la incorporación de plantillas para patrones de integración comunes, las mejores prácticas de la integración se graban en las herramientas, lo que simplifica más todavía las tareas. Cada vez se necesitan menos especialistas con grandes conocimientos de integración y los equipos de aplicación podrían (en teoría) encargarse de algunas integraciones, como veremos en la siguiente sección sobre integración descentralizada.

Los ejecutables de integración moderna están perfectamente adaptados a los tres aspectos de la arquitectura de integración ágil: Implementación detallada, propiedad descentralizada y verdadera infraestructura nativa de nube.

¿Desea un análisis más completo de la infraestructura nativa de nube? [Descargue nuestro libro sobre arquitectura de integración ágil ahora mismo.](#)

## Arquitectura de integración ágil para la plataforma de integración

En este documento, nos hemos centrado en las características de integración de las aplicaciones cuando se las implementa en una arquitectura de integración ágil. Sin embargo, muchas soluciones empresariales se resuelven únicamente a través de la aplicación de capacidades de integración crítica. Una plataforma de integración (o como algunos analistas prefieren llamarla, una “plataforma de integración híbrida”) combina estas capacidades, permitiendo que las organizaciones creen soluciones de forma más uniforme y eficiente.

Muchos especialistas del sector coinciden en el valor que ofrece esta plataforma de integración. Gartner comenta:

La plataforma de integración híbrida (HIP) es un entorno de capacidades de gobernanza e integración local y basado en la nube, que permite que personas con diferentes destrezas técnicas (especialistas de integración y personal no especialista) puedan brindar soporte a una amplia gama de casos de uso de integración... Los líderes de aplicaciones encargados de la integración deben aprovechar el marco de capacidades de la HIP para modernizar las estrategias y la infraestructura de integración, de modo que puedan abordar los casos de uso que van surgiendo para la empresa digital<sup>2</sup>.

Algo clave que Gartner destaca es que la plataforma de integración permite que muchas personas en toda la organización puedan trabajar en las experiencias de usuario que mejor se adapten a sus necesidades. Esto significa que los usuarios comerciales pueden ser productivos en una experiencia más simple que les permite solucionar problemas directos, en tanto que los especialistas de TI tienen niveles de control expertos para superar las situaciones empresariales más complejas. Estos usuarios pueden trabajar juntos mediante la reutilización de los activos que se han compartido, manteniendo la gobernanza del total.

Ocuparse de los casos de uso que van surgiendo de la transformación digital es tan importante como brindar respaldo a las diversas comunidades de usuarios. La mayor parte de este documento se dedicará a explorar esos casos emergentes, pero primero debemos profundizar en las capacidades clave que deben ser parte de la plataforma de integración.

## La plataforma IBM Cloud Integration

IBM Cloud Integration reúne un grupo fundamental de capacidades de integración en una plataforma uniforme que es simple, rápida y confiable. Le permite, en minutos, crear integraciones poderosas y API fácilmente, ofrece desempeño y escalabilidad líderes y ofrece funcionalidades de extremo a extremo sin igual, con seguridad para entornos empresariales.

Dentro de la plataforma IBM Cloud Integration, hemos organizado las seis especialidades de integración clave, cada una la mejor en su clase. Son:

### Gestión de API:

Expone y administra los servicios de la empresa como API reutilizables para ciertas comunidades de desarrolladores, tanto dentro como fuera de su organización. Las organizaciones adoptan una estrategia de API para acelerar la eficacia con la que pueden compartir sus datos exclusivos y activos de servicio, para luego integrarlos en nuevas aplicaciones y oportunidades empresariales.

### Puerta de enlace de seguridad:

Extiende la conectividad y la integración más allá de la empresa con capacidades del perímetro preparadas para DMZ que protegen las API, los datos que mueven y los sistemas subyacentes.

### Integración de aplicaciones:

Conecta las aplicaciones y las fuentes de datos locales o en la nube, para coordinar el intercambio de información del negocio para que los datos estén disponibles cuándo y dónde son necesarios.

<sup>2</sup>Hype Cycle for Application Infrastructure and Integration, 2017, Elizabeth Golluscio.

**Mensajería:**

Asegura la disponibilidad de la información en tiempo real desde cualquier lugar y en cualquier momento, proporcionando mensajes confiables sin pérdida de la información, duplicación ni recuperación compleja en caso de fallas del sistema o la red.

**Integración de datos:**

Accede a los datos, los limpia y los prepara para crear una vista uniforme del negocio dentro de un almacén o lago de datos, para fines analíticos.

**Transferencia a altavelocidad:**

Traslade grandes cantidades de datos entre la nube y sistemas locales, o de nube a nube en forma rápida y predecible, con mayores niveles de seguridad. Facilita la velocidad con que las organizaciones pueden adoptar plataformas en la nube cuando la cantidad de datos es muy grande.

Confiamos que este documento lo haya ayudado a tener una mayor perspectiva de las diversas capacidades cruciales necesarias para una plataforma de integración, más información sobre los requisitos de dichas capacidades para trabajar juntas, y conocimiento sobre cómo la arquitectura de integración ágil puede adoptarse y brindar mayor agilidad, escalabilidad y resiliencia para la plataforma.

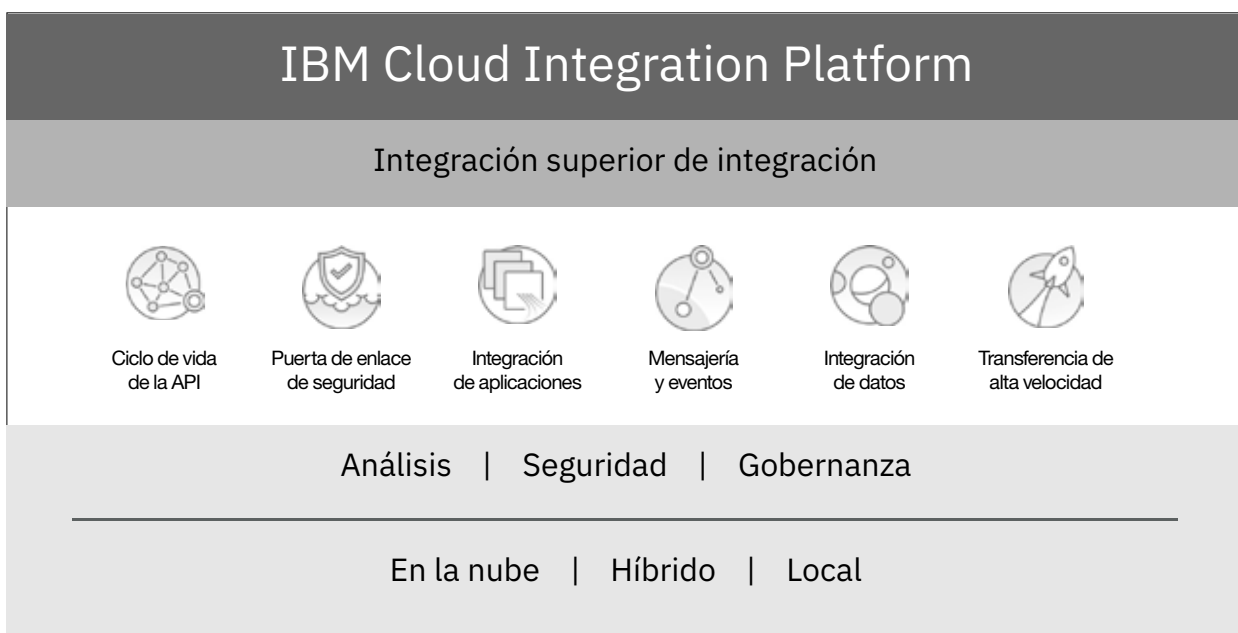


Figura 2: La plataforma IBM Cloud Integration

Asegúrese de descargar el completo [libro electrónico](#) para obtener más información todavía sobre la arquitectura de integración ágil.



© Copyright IBM Corporation 2018

IBM Corporation  
Global Technology Services  
Route 100  
Somers, NY 10589

Producido en los Estados Unidos de América  
Agosto de 2018

IBM, el logotipo de IBM, **ibm.com**, iSeries, Power, System Storage, zEnterprise, TDMF, AIX, BladeCenter y pSeries son marcas comerciales de International Business Machines Corp., registradas en diversas jurisdicciones a nivel mundial. Otros nombres de productos y servicios podrían ser marcas comerciales de IBM o de otras compañías. Hay una lista actualizada de las marcas comerciales de IBM disponible en la Web en "Copyright and trademark information" en [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Linux es una marca comercial registrada de Linus Torvalds en los Estados Unidos, en otros países o ambos.

Microsoft, Windows y Windows NT son marcas comerciales de Microsoft Corporation en los Estados Unidos, otros países o ambos.

El contenido del presente documento (que incluye referencias a monedas o precios sin impuestos) es actual al momento de la publicación e IBM puede cambiarlo en cualquier momento. No todas las ofertas están disponibles en todos los países donde opera IBM.

Los datos de rendimiento y los ejemplos de clientes citados se presentan solo con fines ilustrativos. Los resultados de rendimiento reales pueden variar según las configuraciones específicas y las condiciones operativas.

Es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier otro producto o programa con productos y programas de IBM.

LA INFORMACIÓN DE ESTE DOCUMENTO SE PROPORCIONA "TAL CUAL" SIN GARANTÍAS DE NINGÚN TIPO, YA SEAN EXPRESAS O IMPLÍCITAS, INCLUIDA CUALQUIER GARANTÍA DE COMERCIALIZACIÓN O DE IDONEIDAD PARA UN PROPÓSITO ESPECÍFICO Y CUALQUIER GARANTÍA O CONDICIÓN DE NO VIOLACIÓN. Los productos de IBM están garantizados según los términos y condiciones de los acuerdos conforme a los cuales se brindan.

La capacidad de almacenamiento disponible real informada puede ser para datos sin comprimir y comprimidos, por lo que variará y puede ser menos de la indicada.



Por favor, recicle