

# GUÍA EJECUTIVA PARA ELEGIR UNA PLATAFORMA DE DESARROLLO DE APLICACIONES NATIVAS DE LA NUBE

## INTRODUCCIÓN

Cuando una organización comienza a planificar para aplicaciones nativas de la nube, es importante considerar todo el período de tiempo comprendido: desde la selección de una plataforma de desarrollo hasta que una aplicación sea realmente de nivel productivo y esté preparada para ser distribuida en la nube. Puede ser un largo camino, con muchas decisiones que tomar durante el recorrido que pueden impulsar o dificultar el progreso.

Por ejemplo, al comienzo de la migración a un desarrollo de aplicaciones nativas de la nube, es muy probable que haya ineficiencias si los desarrolladores comienzan a seleccionar herramientas y marcos antes de saber dónde se implementará la aplicación. Mientras los desarrolladores empresariales desean tener opciones de tiempos de ejecución, marcos y lenguajes, las organizaciones necesitan estándares que aborden el ciclo de vida completo de las aplicaciones para reducir los costos operativos, disminuir los riesgos y cumplir con los requerimientos de conformidad. Las organizaciones también desean evitar la dependencia, tanto si es de un solo proveedor de infraestructura de nube o del último estilo de arquitectura.

Además, una vez lograda una curva de aprendizaje muy pronunciada para el desarrollo de nube, las consideraciones para diseñar aplicaciones sólidas, escalables y resistentes no se pueden postergar demasiado en el ciclo de desarrollo. Uno de los factores principales de migrar a la nube es la capacidad flexible para la escalabilidad dinámica, por lo que la resistencia no se puede dejar para lo último.

Para una estrategia efectiva de aplicaciones nativas de la nube, se debe considerar el panorama completo, desde las herramientas de desarrollo y la plataforma de nube hasta la implementación de la automatización y las operaciones. El beneficio de un enfoque holístico para el desarrollo de aplicaciones nativas de la nube es prestarse para un recorrido guiado que elimina las desviaciones innecesarias.

## EL CONCEPTO DE DESARROLLO DE APLICACIONES NATIVAS DE LA NUBE

El desarrollo de aplicaciones nativas de la nube implica responder al cambio con velocidad, resistencia y agilidad. Esta respuesta se logra mediante implementaciones más frecuentes que reducen significativamente los tiempos de entrega para responder al cambio.

Gran parte de esta discusión sobre las aplicaciones nativas de la nube se remonta a un manifiesto llamado “[The Twelve-Factor App](#)” (La aplicación de doce factores), que se trata de un conjunto de principios recogidos de la experiencia obtenida durante el diseño y la ejecución de aplicaciones de software como servicio (SaaS). Los objetivos son:

- Reducir la curva de aprendizaje del desarrollador.
- Crear aplicaciones que se adapten bien a la implementación de plataformas de nube.
- Maximizar la agilidad a través de la implementación continua.
- Habilitar aplicaciones para escalar verticalmente sin necesidad de cambios significativos.



facebook.com/redhatinc  
@RedHatIberia  
Red Hat EMEA

Trabajando en conjunto con los clientes, Red Hat ha descubierto los siguientes elementos clave para el desarrollo de aplicaciones que son nativas de la nube:

- **Una arquitectura basada en servicios.** El estilo puede ser de microservicios o de algún modelo de arquitectura modular y sin conexión directa. Los servicios representan una actividad comercial que es bastante autónoma y se comprende fácilmente. El objetivo es tener servicios que se puedan actualizar o reemplazar fácilmente. Un servicio bien definido es más fácil de probar en detalle que una aplicación completa.
- **Contenedores.** Los contenedores Linux®, que usan imágenes de Docker, son un modelo de paquete común y un entorno de ejecución autónoma que proporciona portabilidad y aislamiento. Los contenedores permiten la automatización avanzada que hace tan atractivas a las plataformas de nube.
- **Automatización DevOps.** Un conjunto de procesos y prácticas de colaboración destinados a unificar el desarrollo y las operaciones. Los objetivos son mejorar la frecuencia de implementación y ofrecer lanzamientos de mejor calidad, lo que da como resultado tiempos más rápidos para comercializar, menores riesgos y mayor satisfacción del usuario. La integración y la distribución continuas (CI/CD) están estrechamente relacionadas con las mejoras de DevOps. Otro objetivo clave además de garantizar una experiencia de usuario final de calidad es la instrumentación y el control para comprender el rendimiento.
- **Comunicación basada en API.** La comunicación entre procesos solo ocurre a través de las interfaces de programación de aplicaciones (API), mediante el uso de interfaces limpias y basadas en contratos a través de la red. Esto elimina el acoplamiento inesperado que restringe el cambio y que es una fuente común de interrupciones. Esto es de especial importancia en los entornos híbridos, donde las aplicaciones ya no residen en el mismo centro de datos.

## ¿QUÉ SON LOS MICROSERVICIOS?

La arquitectura de microservicios es un estilo de arquitectura atractivo para resolver numerosos desafíos de desarrollo de software. Las aplicaciones se desglosan en un conjunto de servicios sin conexión directa que implementan funciones comerciales específicas. El objetivo principal es la innovación más rápida con base en la idea de que un microservicio individual es mucho más fácil de comprender, mejorar, probar e implementar que una aplicación monolítica. Como el propósito y la función de cada microservicio deben estar bien definidos, la prueba automática y la distribución continua resultan mucho más prácticas para implementar.

## REQUERIMIENTOS PARA LAS APLICACIONES NATIVAS DE LA NUBE

La arquitectura nativa de la nube coloca numerosos requerimientos técnicos en las aplicaciones. Algunos de estos son:

- **Necesidades de implementación que se deben automatizar.** Para acelerar la frecuencia de lanzamientos es necesario incorporar la implementación automatizada para que el proceso pueda ocurrir simplemente haciendo clic en un botón para su aprobación. Se necesita un canal automatizado de diseño e implementación junto con la capacidad para revertir rápidamente a un lanzamiento anterior, en caso de que fuera necesario.
- **La información de configuración debe estar separada de la aplicación.** Los detalles de configuración necesarios para ejecutar una aplicación en un entorno específico se deben almacenar en el entorno de implementación. Esto permite que se use la misma imagen de la aplicación en todos los entornos, incluidos el desarrollo, las pruebas y la producción. Además, el almacenamiento seguro de credenciales es necesario, ya que las aplicaciones necesitan comunicarse de forma segura con las fuentes de datos y otros componentes a través de la red.

- **Los servicios de aplicaciones se deben localizar de forma dinámica.** Esto es necesario para la portabilidad en los entornos, la alta disponibilidad y la escalabilidad dinámica, donde se usa el equilibrio de cargas.
- **Se necesitan almacenes de datos separados para conservar los datos.** Los procesos y contenedores no deben tener estado. Los procesos se pueden caer y ser restablecidos, posiblemente en máquinas distintas, en unos pocos segundos. Los datos, la información de la sesión y los registros se deben mantener en almacenes de datos externos.

Algunas consideraciones a tener en cuenta al comienzo del desarrollo de aplicaciones nativas de la nube son:

- ¿Cuánta experiencia tiene su organización en el diseño y la implementación de aplicaciones en contenedores para ejecutarlas en plataformas de nube? ¿Comprenden los desarrolladores qué se necesita para gestionar datos persistentes y la configuración de forma separada de las aplicaciones?
- ¿Cuánto tiempo implica el diseño de un entorno de desarrollo para comenzar el desarrollo de aplicaciones nativas de la nube? ¿Coincide el entorno de desarrollo con el entorno de nube de producción?
- ¿Cuánto tiempo tarda diseñar e integrar eficazmente CI/CD en los procesos de desarrollo e implementación?

Todos estos factores se deben tener en cuenta al seleccionar una plataforma de desarrollo de aplicaciones nativas de la nube.

## CONSIDERACIONES DE LOS MICROSERVICIOS

Los microservicios y las aplicaciones nativas de la nube están estrechamente relacionados. De hecho, los microservicios son difíciles de implementar con éxito sin una plataforma de nube, contenedores y automatización DevOps. Uno de los factores principales para los microservicios es la agilidad que se obtiene al usar componentes de aplicaciones más pequeños que se pueden lanzar con más frecuencia y menos riesgos. Sin embargo, lo que antes era una sola aplicación puede convertirse en varios microservicios con ciclos de lanzamiento independientes. Las siguientes son algunas consideraciones a la hora de migrar a los microservicios:

- ¿Cuánto tiempo implica aprovisionar nuevos entornos de desarrollo, pruebas y producción para cada nuevo microservicio?
- ¿El diseño y la implementación están lo suficientemente automatizados como para dar soporte a la cantidad de lanzamientos que se necesitarán?
- ¿Cómo detectarán las aplicaciones dónde se ejecutan los microservicios? ¿Se cuenta con la flexibilidad suficiente como para reubicar los microservicios, según sea necesario, para su disponibilidad o escalabilidad?
- ¿Se pueden lanzar los microservicios sin incurrir en interrupciones del servicio que impacten en los usuarios finales?
- ¿Qué tan difícil es diagnosticar los problemas? ¿Qué tan difícil es rastrear los eventos en diferentes servicios y, posiblemente, en entornos híbridos?
- ¿Los problemas con un servicio estarán bien contenidos o esos problemas se convertirán en múltiples fallas?
- Dada la naturaleza distribuida de los microservicios, los componentes adicionales de infraestructura de software se necesitan para realizar funciones tales como la detección de servicios, el equilibrio de carga del lado del cliente, la gestión de estado persistente, el rastreo distribuido y la resiliencia.

- ¿Necesitarán los desarrolladores implementar esos servicios o encontrarán componentes fuera de Internet?
- ¿Cuánta carga de mantenimiento habrá si estos componentes ahora son parte de cada microservicio implementado? ¿Cómo se actualizarán los componentes para abordar los errores y las vulnerabilidades de seguridad?
- ¿Son estos servicios básicos que deberían integrarse en plataformas o tiempos de ejecución de la nube?

Una de las primeras preguntas que se hacen los desarrolladores al comenzar a diseñar microservicios es qué tipo de tiempos de ejecución, marcos y lenguajes son necesarios. Algunas consideraciones al responder esta pregunta son:

- ¿Necesitan los desarrolladores conocer nuevos marcos para poder comenzar?
- ¿Se pueden aprovechar el código Java™ EE y la experiencia de la organización al migrar a los microservicios?
- ¿Se pueden usar los nuevos estilos de desarrollo, como la programación reactiva? ¿Ofrece la plataforma las opciones necesarias para satisfacer las nuevas demandas, como las cargas de trabajo basadas en eventos y creadas por la tecnología móvil y el Internet de las cosas (IoT)?
- Con el fin de usar la mejor herramienta para el trabajo, ¿es razonable tener distintos microservicios implementados con diferentes tiempos de ejecución, marcos o lenguajes? ¿Habrá diferencias en la forma en la que están configurados, implementados o protegidos?
- ¿Cuánto tiempo transcurrirá hasta que los nuevos microservicios se puedan probar en un entorno de nube pública real?

Los microservicios son muy interesantes. Sin embargo, los microservicios y las arquitecturas de aplicaciones distribuidas pueden ser difíciles de dominar. Es importante seleccionar una plataforma que pueda reducir la complejidad y simplificar el desarrollo.

### **MODERNIZAR LAS APLICACIONES MONOLÍTICAS EXISTENTES**

Si bien puede ser atractivo para los desarrolladores, no siempre es práctico o rentable rescribir las aplicaciones monolíticas existentes en microservicios. Sin embargo, estas aplicaciones aún deben ser mantenidas y tener una acumulación de mejoras que deben ser implementadas. Reducir la acumulación puede ser complejo debido a la longitud de los ciclos de lanzamiento. Realocar estas aplicaciones en una plataforma de nube puede facilitar la implementación de CI/CD, junto con lanzamientos graduales, como las implementaciones azul/verde o "canary". Esto permite a los desarrolladores proporcionar lanzamientos más frecuentes con menos riesgos. La aplicación alojada en la nube puede ser una "aplicación monolítica veloz", lo que permite a los desarrolladores abordar la acumulación de mejoras.

Realocar una aplicación en la nube también puede ser un primer paso útil para implementar microservicios. La flexibilidad de la nube facilita la implementación de nuevos microservicios en contenedores junto con las aplicaciones existentes. Los desarrolladores pueden implementar entonces los microservicios que migran la funcionalidad fuera de la aplicación monolítica.

Dados los beneficios de cargar y mover las aplicaciones existentes a la nube, estas son algunas consideraciones a la hora de elegir una plataforma de desarrollo de aplicaciones nativas de la nube:

- ¿Tiene la plataforma de desarrollo opciones para modernizar las aplicaciones existentes, así como nuevos desarrollos totalmente nuevos?
- ¿Es compatible la plataforma de nube con las aplicaciones que todavía no son totalmente nativas de la nube?

- ¿Qué opciones hay para migrar las aplicaciones de Java EE existentes?
- ¿Permite la plataforma de desarrollo la aceleración de la distribución de aplicaciones de estilo monolítico?

Los servidores de las aplicaciones pueden ser un entorno de tiempo de ejecución viable para las aplicaciones nativas de la nube, pero la función del servidor de aplicaciones tradicional cambia al migrar a una plataforma nativa de la nube. Tradicionalmente, los servidores de aplicaciones proporcionaban un entorno de implementación y tiempo de ejecución, así como servicios de gestión para un dominio central ejecutado en un clúster de sistemas. Sin embargo, muchas de las herramientas operativas incluidas, como la consola de administración, ya no son necesarias y son contraproducentes. Las plataformas de nube son atractivas porque incluyen capacidades de gestión automatizadas que permiten escalabilidad dinámica y distribución continua. Empaquetar un servidor de aplicaciones tradicional con sus componentes de gestión en un contenedor puede impedir que los desarrolladores aprovechen esas capacidades. Algunas consideraciones a tener en cuenta:

- ¿Incluye la plataforma una opción para un tiempo de ejecución Java EE que esté integrado con las capacidades de gestión de la plataforma de nube?
- ¿Hay herramientas que ayuden a migrar las aplicaciones que se ejecutan en un servidor de aplicaciones heredado a un entorno Java EE moderno y basado en la nube?
- Para las aplicaciones que no usan todas las capacidades de Java EE, como las aplicaciones web, ¿hay alternativas que puedan adaptarse mejor a la nube?

Con la plataforma y las herramientas correctas, las organizaciones pueden obtener más valor de las aplicaciones existentes y migrar a un modelo nativo de la nube y de microservicios.

### COMPATIBLE LOCALMENTE Y CON NUBES MÚLTIPLES

La mayoría de las organizaciones de TI prefieren no estar limitadas a un solo proveedor de infraestructura de nube pública. Además, muchas organizaciones creen que, incluso en años, al menos la mitad de sus aplicaciones seguirán ejecutándose in situ. Debido a que la mayoría de las organizaciones tienen estos dos requerimientos, algunas consideraciones son las siguientes:

- ¿Deberán cambiar las aplicaciones a fin de usar la infraestructura de nube de distintos proveedores? ¿Qué sucede con la implementación, las operaciones y la supervisión? ¿Serán diferentes en cada plataforma de nube?
- ¿Cómo se pueden minimizar las diferencias entre la infraestructura de nube pública y los sistemas in situ? ¿Se usa la plataforma de nube en nubes públicas también disponibles para ejecutarse in situ o en la infraestructura como servicio (IaaS) preferida de la organización?

### UN PANORAMA MÁS AMPLIO

Al evaluar qué plataforma de desarrollo de aplicaciones nativas de la nube elegir, también vale considerar cómo impactará esta elección en otras áreas:

- ¿Existen herramientas de desarrollo disponibles que funcionen con el desarrollo de la nube y la plataforma de implementación desde un comienzo? Si no es así, ¿cuánto tiempo invertirán los desarrolladores tratando de configurar e integrar sus herramientas?
- ¿Proporciona la plataforma a los desarrolladores un enfoque prescriptivo o guiado para incrementar la productividad?
- ¿Existe un conjunto de servicios de middleware basados en la nube para la mensajería, el almacenamiento de datos y la gestión de reglas o procesos empresariales disponibles y listos para ejecutarse en la plataforma cloud?

- ¿Están disponibles los contenedores prediseñados de terceros para integrarse con las aplicaciones?
- ¿Están disponibles los servicios de capacitación y consultoría? ¿Tienen los recursos internos disponibles experiencia en modernización de aplicaciones?

### **LA IMPORTANCIA DE ELEGIR LA PLATAFORMA DE DESARROLLO PARA LA NUBE CORRECTA**

Para los desarrolladores que recién comienzan con el desarrollo en la nube es natural resolver un problema a la vez, utilizando los tutoriales y eligiendo los componentes de software de Internet. Primero, deben aprender cómo ensamblar lo suficiente de un entorno de tiempo de ejecución para crear un código de aplicación en un contenedor. Luego, deben decidir qué otros componentes de software son necesarios para crear aplicaciones completas. Este proceso puede ser lento y presentar muchos inconvenientes:

- La curva de aprendizaje y la cantidad de integración requerida pueden causar una pérdida de productividad del desarrollador, que es difícil de explicar a la empresa. Todos los nuevos desarrolladores contratados deberán conocer la pila que ha sido creada localmente.
- Todo componente de software seleccionado deberá ser mantenido para asegurarse de que no contenga errores conocidos ni vulnerabilidades de seguridad. También surge la duda de si los componentes de open source son apropiados para el uso empresarial.
- Muchas organizaciones informan que los desarrolladores pueden terminar escribiendo códigos para funciones, como la configuración y la implementación, que no proporcionan ningún valor empresarial directo. Peor aún, muchas de estas funciones están sujetas a la opción de una plataforma de implementación y son servicios básicos que deben ser parte de la plataforma de implementación.

Tal vez el problema más grande con el ensamblado de una colección de componentes es que no se presta a oportunidades para reducir la curva general de aprendizaje para el desarrollo y las operaciones. La alternativa es elegir una plataforma de desarrollo de nube que esté diseñada para manejar todas las etapas, desde seleccionar un tiempo de ejecución para comenzar el desarrollo y todo el recorrido hasta lograr la implementación de producción.

Migrar a aplicaciones nativas de la nube es un proceso que no ocurre en una noche. Para muchas organizaciones, es un proceso que mejora con la experiencia. Dado todo lo que implica el desarrollo de las aplicaciones nativas de la nube, está claro que un enfoque integral que comprenda desde el desarrollo hasta la implementación es más factible de ser exitoso. Red Hat ofrece una plataforma así con Red Hat® OpenShift Application Runtimes y Red Hat OpenShift.

### **RED HAT OPENSIFT APPLICATION RUNTIMES Y RED HAT OPENSIFT**

Red Hat OpenShift Application Runtimes está diseñada para simplificar el desarrollo de las aplicaciones nativas de la nube. Un conjunto organizado de tiempos de ejecución y marcos integrados proporciona un camino guiado con una experiencia prescriptiva que se puede usar para poner en marcha el desarrollo. Es posible contar con una plataforma de desarrollo e implementación totalmente optimizada porque OpenShift Application Runtimes se basa en –y está optimizada para– Red Hat OpenShift, una plataforma de aplicaciones en contenedor que está diseñada para las nubes híbridas.

Red Hat OpenShift proporciona una plataforma de autoservicio para los desarrolladores y las operaciones con el fin de crear y ejecutar aplicaciones en contenedores. Mediante OpenShift, se puede aprovisionar en minutos un entorno para un nuevo microservicio o una aplicación. Con OpenShift, también es posible reducir significativamente los ciclos de implementación con muchos menos riesgos. Y con solo hacer unos clics, se dispone de canales de implementación y

diseño de CI/CD automatizados y potentes. Juntos, Red Hat OpenShift Application Runtimes y Red Hat OpenShift, con las herramientas de desarrollo de Red Hat y Red Hat Consulting, pueden ayudar a las organizaciones a migrar a aplicaciones nativas de la nube en menos tiempo y con menos riesgos.

Los tiempos de ejecución en OpenShift Application Runtimes se seleccionan para darles a los desarrolladores la herramienta correcta para el trabajo. Para el desarrollo de microservicios, los desarrolladores tienen la opción de usar Java EE para aprovechar la experiencia existente, el estándar más reciente de Microprofile de Java que ha evolucionado para satisfacer las necesidades de los microservicios o un marco basado en eventos para diseñar microservicios reactivos que pueden escalar para cargas de trabajo de gran ocurrencia y baja latencia. Se proporciona un tiempo de ejecución Node.js para los servicios JavaScript de backend que frecuentemente se usan con aplicaciones móviles y web. Para migrar las aplicaciones existentes, se proporciona un tiempo de ejecución basado en Red Hat JBoss® Enterprise Application Platform, un servidor de aplicaciones Java EE 7 certificado basado en una arquitectura moderna, modular y lista para la nube. Todos los tiempos de ejecución están probados y verificados por Red Hat.

Para que los desarrolladores comiencen rápidamente, está disponible Launch, una herramienta de software como servicio gratis y basada en la nube que se ejecuta en Red Hat OpenShift Online, en el sitio web de Red Hat Developers. Después de elegir una aplicación y un tiempo de ejecución como ejemplos, los desarrolladores obtienen una base de códigos completa que está lista para diseñar y ejecutar en la nube en OpenShift Online. Como herramienta educativa, las mismas aplicaciones de ejemplo están disponibles para todos los tiempos de ejecución, lo que permite que los desarrolladores comparen fácilmente los méritos de los estilos de arquitectura disponibles.

Para obtener más información visite: <https://www.redhat.com/es/technologies/cloud-computing/openshift/application-runtimes>



## ACERCA DE RED HAT, INC.

Red Hat es el proveedor líder mundial de soluciones open source empresarial, con un enfoque impulsado por la comunidad para la obtención de tecnologías cloud, Linux, middleware, almacenamiento y virtualización de alta fiabilidad y rendimiento. Red Hat también ofrece servicios de soporte, formación y consultoría. Como eje central de una red global de empresas, partners y comunidades open source, Red Hat ayuda a crear tecnologías competentes e innovadoras que liberan recursos para el crecimiento y preparación de los consumidores para el futuro de las TI. Conozca más en <http://es.redhat.com>.

### ARGENTINA

Ingeniero Butty 240, 14º piso  
Ciudad de Buenos Aires  
Argentina  
+54 11 4329 7300

### CHILE

Avda. Apoquindo N° 2827  
oficina 701, Piso 7  
Los Condes, Santiago, Chile  
+562 2597 7000

### COLOMBIA

Red Hat Colombia S.A.S  
Cra 9 No. 115-06 Piso 19 Of 1906  
Edificio Tierra Firme Bogota, Colombia  
+571 5088631  
+52 55 8851 6400

### MÉXICO

Calle Río Lerma 232  
Cuauhtémoc  
06500 Ciudad de México  
Mexico  
+52 55 8851 6400

### ESPAÑA

Torre de Cristal  
Paseo de la Castellana 259C  
Piso 17 Norte  
28046 Madrid  
+34 914148800



[facebook.com/redhatinc](https://facebook.com/redhatinc)  
@RedHatIberia  
Red Hat EMEA

[es.redhat.com](https://es.redhat.com)  
F10657\_V1\_0118\_KVM

Copyright © 2018 Red Hat, Inc. Red Hat, Red Hat Enterprise Linux, el logotipo de Shadowman y JBoss son marcas comerciales de Red Hat, Inc. registradas en Estados Unidos y en otros países. Linux® es la marca comercial registrada de Linus Torvalds en EE. UU. y en otros países. Java y todas las marcas comerciales y los logotipos basados en Java son marcas comerciales o marcas comerciales registradas de Oracle America, Inc. en los Estados Unidos y en otros países.