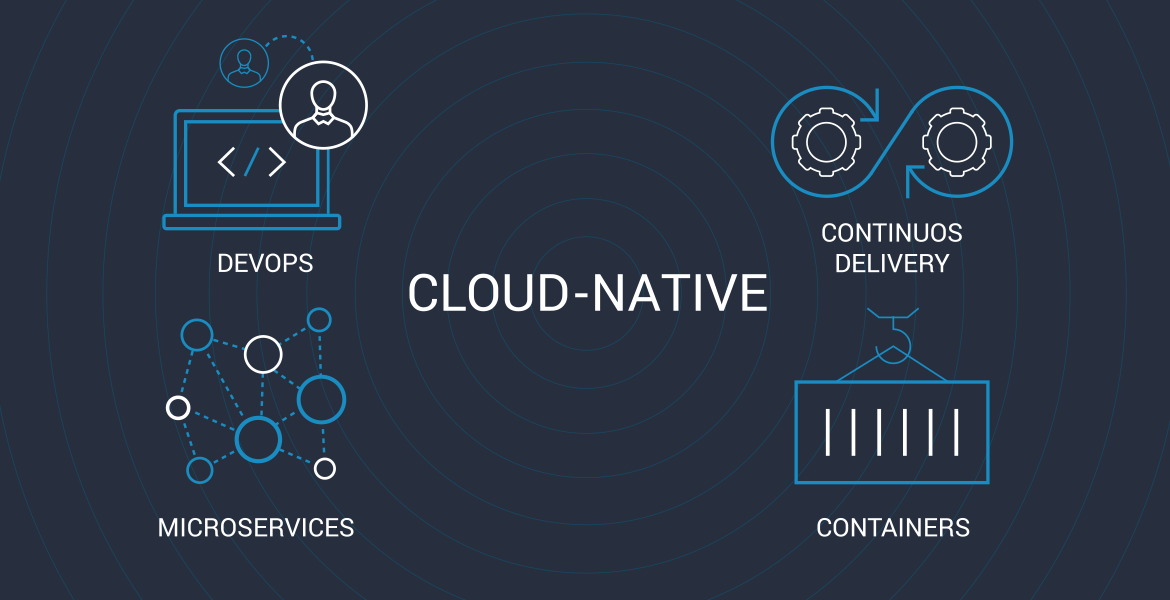
Cloud Native – principios, aplicaciones y desafíos



¿Qué es Cloud Native?

Cloud Native (nube nativa) es un patrón de arquitectura de software para desarrollar aplicaciones usando principios esenciales de cloud computing como la escalabilidad, elasticidad y agilidad. En este artículo exploramos las bases de Cloud Native para **crear, distribuir y ejecutar las aplicaciones usando lo mejor de la tecnología cloud**.

Objetivos de Cloud Native

Las arquitecturas nativas cloud están diseñadas para mejorar aplicaciones tradicionales o hacer nuevos desarrollos con valores muy simples: **más rápido, de mejor calidad y por un costo menor**.

Los beneficios de Cloud Native se reflejan en sus principios:

Escalabilidad

Posibilidad de que los recursos y aplicaciones escalen horizontal o verticalmente de manera rápida y dinámica, con pocas restricciones en términos del hardware en que se ejecutan.

Velocidad

Desarrollar, probar y entregar código de calidad mucho más rápido. Cloud Native aumenta la capacidad de los equipos sin aumentar los recursos de las empresas.

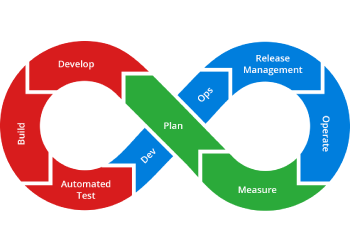
Costos Marginales

Desde un punto estratégico, Cloud Native aprovecha la facturación flexible de proveedores cloud así como el dinamismo de pagar por recursos adicionales solo cuando sean necesarios. En entornos on-premise, permite agregar o quitar hardware de manera dinámica sin afectar el funcionamiento base del sistema.

Aplicaciones de Cloud Native

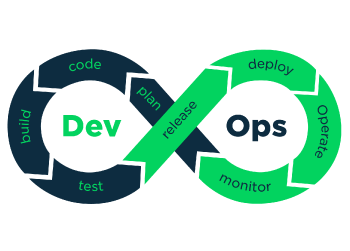
El enfoque Cloud Native permite **moverse rápidamente, con pasos pequeños, reversibles y reduciendo el riesgo técnico**. Este ideal no es fácil de obtener, ni tampoco es gratis. Implica un cambio importante en la cultura y filosofía de desarrollo así como múltiples retos técnicos.

Continuous delivery, DevOps,  Microservicios y Contenedores: los cuatro pilares de Cloud Native



Continuous Delivery (entrega continua)

Continuous Delivery es la motivación base para cloud-native: entregar software más rápido, recibiendo retroalimentación en menor tiempo. Esta práctica nos permite a los desarrolladores crear, probar y preparar automáticamente cambios en código, verificando actualizaciones en las aplicaciones en varias dimensiones antes de enviarlas a sus usuarios. Todo esto con el objetivo de mejorar la eficacia y rapidez de la entrega de software

DevOps

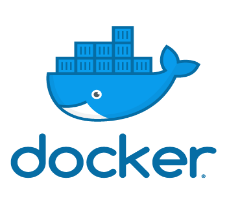
[DevOps](https://www.aplyca.com/es/devops-cloud) es la manera en que afrontamos los retos culturales y técnicos para una estrategia cloud-native. La palabra combina las prácticas de desarrollo (Dev) con operaciones (Ops) y permite adaptarse al entorno de Cloud Native para asegurar que una empresa entregue constantemente un software de calidad – y al mismo tiempo – maximice el retorno de inversión.  A medida que evoluciona el negocio también lo hacen sus prácticas DevOps.

Microservicios

Es el patrón de arquitectura usado para efectivamente expandir el proceso de desarrollo y entrega de software, evitando las estrategias lentas y riesgos de los desarrollos monolíticos. Con Microservicios**desarrollamos nuestra aplicación en pequeños servicios de manera totalmente independiente**.

Al ser independientes pueden usar diferentes lenguajes o plataformas, ser liderados por equipos distintos y en general ser mucho más agresivos para lanzar nuevas funcionalidades sin afectar las demás. Interactúan entre sí a través de la exposición de API's y mantienen su independencia. Una de las ventajas de usar Microservicios es que adaptan su escala fácilmente a la demanda del sistema.

Contenedores

En su esencia los contenedores son una manera de **empaquetar software.**Un contenedor se ejecuta de manera predecible, repetible e inmutable. No hay errores asociados a mover el código de máquina en máquina, pues al lanzar nueva funcionalidad, se despliega**máquina y código en un mismo contenedor.**Todo lo que el servicio necesita, las librerías, dependencias, sistema operativo está empaquetado en el contenedor en sí. Es similar a trabajar con máquinas virtuales, pero sin la carga e ineficiencia de montar un sistema operativo completo.

Un contenedor es una especie de máquina virtual muy portable y menos exigente en cuanto a recursos de cómputo que las máquinas virtuales convencionales (por eso se las llama livianas o ligeras).

Los contenedores son entonces un herramienta fantástica para iniciativas Cloud Native modernas. Pueden ejecutar microservicios, funciones independientes o incluso aplicaciones monolíticas tradicionales.

Nubes privadas, públicas o híbridas

Un atractivo importante de Cloud Native es no requerir un compromiso de dónde se alojan estas aplicaciones. Puede hacerse en diferentes operadores cloud como Amazon Web Services, Azure o Google Cloud Platform, o combinarlo con nubes privadas on-premise. Cloud Native no se limita a grandes empresas, sino que permite que pequeñas empresas –con el suficiente compromiso– sean más competitivas.

El uso efectivo de tecnologías de orquestación como [Kubernetes](https://kubernetes.io/" \t "_blank), PaaS ([Open Shift](https://www.openshift.com/)) o herramientas de automatización como RedHat [Ansible](https://www.redhat.com/es/technologies/management/ansible) y [Terraform](https://www.terraform.io/" \t "_blank) empiezan a desdibujar la línea que separa ambientes locales (on-premise) o remotos, entornos corriendo sobre virtualización o hardware directo (*bare-metal*) o las limitaciones de entornos de desarrollo diferentes a producción.

CNCF - Cloud Native Computing Foundation



La importancia de Cloud Native es tan grande que se ha creado un entramado organizacional para garantizar que las decisiones de las decenas de organizaciones interesadas se toman ágilmente y con transparencia.

La fundación [CNCF](https://www.cncf.io/) (Cloud Native Computing Foundation) hace parte de la Linux Foundation y cuenta entre sus [miembros](https://www.cncf.io/about/members/) no solo a los proveedores cloud más importantes como AWS, Azure, Google Cloud sino a decenas de empresas comprometidas con la iniciativa.

Entre sus [proyectos](https://www.cncf.io/projects/) más destacados está el de fomentar un ecosistema para el desarrollo de componentes claves para la iniciativa como [Kubernetes](https://kubernetes.io/) (orquestación), [Prometheus](https://prometheus.io/) (monitoreo), [FluentD](https://www.fluentd.org/) (logueo), [ContainerD](https://containerd.io/) y [rkt](https://coreos.com/rkt/) (ejecución de contenedores) o CNI (API de redes) entre otros. El resultado es que la fundación le da transparencia a tecnologías críticas para el futuro de la computación, sin el sesgo natural de intereses de un solo fabricante. Es así como la CNCF lidera algunos los [proyectos de desarrollo más rápido](https://www.cncf.io/blog/2017/06/05/30-highest-velocity-open-source-projects/)en la historia del código libre (open-source).

Desafíos y consideraciones

El desarrollo de soluciones basadas en Cloud Native se engloba dentro del marco de profundos cambios en el paradigma desarrollo de software de la industria. **La transformación digital exige mejores prácticas en todo el ciclo de desarrollo de software.**

Las aplicaciones Cloud Native son intencionalmente desarrolladas para un modelo cloud. Son construidas –con un ritmo rápido de desarrollo– por **equipos pequeños enfocados a cada tipo de funcionalidad**garantizando escalabilidad, portabilidad entre ambientes, resiliencia y agilidad.

Los equipos de infraestructura pasan a ser impulsores de cambio y valor de negocio

Es importante entender que la **operación de sistemas va a cambiar en un mundo Cloud Native**. Las personas de infraestructura y operaciones pasan de tener un rol más pasivo, de mantenimiento y protección contra incidentes a ser impulsores de procesos de cambio, automatización y generación de valor de negocio.

No todas las aplicaciones tienen que convertirse a Cloud Native, y es algo que los equipos de negocios y TI deben acordar para **darle prioridad a las cargas de trabajo considerando su viabilidad técnica, importancia estratégica y retorno de la inversión (ROI)** caso por caso. En adición a los desarrollos nuevos, en donde tenemos mucha libertad para elegir cómo hacerlos, los retos de actualizar plataformas existentes requieren procesos de cambio más sofisticados a nivel organizacional.

Ref.: <https://www.aplyca.com/es/blog/cloud-native>