

[numerica.latina.red](https://numerica.latina.red)

# "Agnóstica y migrante".- — NUMERICA LATINA

*Beto*

4-5 minutos

---

(Numérica no es una servidora, es una **metodología**.)

Algunas de sus características son que es «**agnóstica y migrante**», esto puede ser un aporte a la seguridad de nuestras infraestructuras, y a la resiliencia de nuestros proyectos... veremos cómo.

## **Agnóstica respecto al alojamiento**

Es como cuando se viaja, se sabe que debemos andar nuestra propia bolsa de dormir. Esta metáfora se refiere a lo que en computación llamaron «**contenedores**», o en inglés *containers*.

Con una arquitectura basada en contenedores, podemos trabajar sobre una plataforma que funciona igual, **independiente de sea cuál sea el sistema que la aloja**. Esto es lo que se llama en informática (y en inglés) «*host-agnostic*».

## **Contenedores y Máquina Virtual**

Pero antes de irnos a la teoría repasemos algunos conceptos.

Una [máquina virtual](#) (o **VM**, por sus siglas en inglés) es un programa que *simula una computadora completa*. Así en una misma máquina 'de fierro', se pueden abrir varios sistemas operativos

simultáneamente.

Un [container](#) es como una VM, pero reducida a los programas que un único servicio necesita. Si una máquina virtual trae *un sistema operativo completo* (por ej. algún sabor de Linux, etc.), **un contenedor trae solamente las librerías necesarias para hacer funcionar un servicio** en específico (por ej. un sitio web en WordPress).

## Mayor seguridad

Si una máquina virtual fuera una casa, con una arquitectura basada en contenedores, *estaríamos separándola en dormitorios*, que solamente comunican entre sí por puertas específicas (bueno, en este caso puertos).

En términos de seguridad, **esto equivale a tener paredes**. En caso de una vulneración, el atacante gana acceso solamente a una habitación\*.

## Automatización

Pero volvamos a nuestra metodología, ya dijimos que la *arquitectura está basada en contenedores...*

Además, combinamos un segundo ingrediente, **la automatización del despliegue de estos servicios**.

Es decir que basta que **definamos**, por ejemplo, que en el dominio [nube.numerica.cl](https://nube.numerica.cl) serviremos un [Nextcloud](#), y **ejecutemos** la orden de despliegue, *para que esto se configure y se levante solo, en cuestión de minutos*.

## Nomadismo virtual

Esto nos da libertad. Siguiendo con la metáfora, *nuestra información es libre de cambiarse de casa, de ciudad, de país*, se vuelve virtualmente migrante.

De hecho el proceso de cambiarse de un alojamiento a otro lleva el mismo nombre que la peor parte de un viaje: la migra.

## Migraciones

Las **migraciones** y los **respaldos** son algunas de las tareas *más técnicas de la administración de sistemas*, y por ende de mayor dependencia. Además, ¿qué hago con un respaldo, cómo lo vuelvo a colocar en Internet?

Esta metodología automatiza también las tareas de respaldo de los datos de una aplicación (como bases de datos, archivos multimedia, etc) con el procedimiento **back**. Su par, el procedimiento inverso **up** actualiza una aplicación con los datos respaldados antes.

Esto nos permite *restaurar a un estado* guardado de esta aplicación, en caso de avería (cosa frecuente, por diversos motivos).

## Back + Up = Migración

Además, se puede usar para desplegar este estado *a una nueva locación* de la aplicación. O sea, resulta **trivial migrar nuestros servicios** de un proveedor de alojamiento a otro, cuando queramos.

---

## Infraestructura como código

Repasando, esta combinación de empaquetado de aplicaciones en contenedores, junto con la automatización de su despliegue, componen una verdadera capa lógica, que vuelve nuestra infraestructura independiente del proveedor subyacente.

Como pueden imaginar, esta independencia otorga resiliencia, pues *en caso de interrupción de nuestros servicios (sea por problemas técnicos, ataques, censura, etc.) facilitan las acciones de restaurarlos, moverlos, etc.*

---

Para saber más acerca de cómo funciona esta metodología en la práctica, aquí profundizamos algunas [ideas](#), y aquí se le puede dar un vistazo al [código](#).