

# EstandaresComputacionNube1.pdf



**Sergio\_Beryuza**



**Legislación y Estandarización**



**3º Grado en Ingeniería Informática**



**Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba**



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.



# blog.trendmicro.es



« [proxima](#)  
[última](#) »

## Estándares de la computación en nube: expectativas frente a realidad

Artículo original de [Justin Foster \(Software Architect, Trend Micro\)](#)

Aunque parezca que la portabilidad e interoperabilidad de la computación en la nube son cuestiones que no afectan a la seguridad, evitar la dependencia absoluta de un solo proveedor implica algo más que tener acceso a precios competitivos o a un mejor servicio. Contar con un solo proveedor supone un riesgo, especialmente en lo que concierne a la disponibilidad de servicios y datos.

A lo largo de los años, la necesidad de portabilidad e interoperabilidad se ha solucionado mediante la estandarización. Así por ejemplo, **la estandarización de las vías férreas ha permitido viajar entre continentes**, al igual que el **protocolo TCP/IP** ha hecho posible las comunicaciones a nivel mundial. No sorprende, por tanto, que un gran número de personas quiera utilizar la computación en nube y que crea en la necesidad de estándares para prevenir la dependencia absoluta de un solo proveedor. Pero, ¿es realmente necesaria la adopción general de estándares? Lo cierto es que, aunque no se trate de un método ideal, esto **permitiría la interoperabilidad mediante abstracción (o intermediación) y la portabilidad mediante la conversión en un entorno con muchos estándares.**

Cuando se habla de la interoperabilidad y portabilidad de una infraestructura como servicio (IaaS), generalmente hay dos cuestiones importantes. Una es el formato de las plantillas de los equipos virtuales (o imágenes) que describe el disco y la configuración de los recursos virtuales requeridos. Si bien es la solución de virtualización subyacente utilizada la que lo determina, algunos proveedores han creado formatos personalizados (por ejemplo, Amazon Machine Image). **El Formato de visualización abierto (OVF)** se ha diseñado como estándar único, pero **los proveedores públicos seguirán promocionando sus distintos formatos por diversas razones.** Sin la adopción global de OVF, la siguiente opción sería la conversión del formato para ofrecer una portabilidad viable. Como recurso provisional, algunos proveedores de servicios han comenzado a aceptar múltiples formatos para **evitar la sobrecarga a causa de las conversiones**, del mismo modo que algunos dispositivos aceptaron los formatos **HD DVD y Blu Ray** hasta que se ganó la batalla de los estándares.



El otro reto es la incompatibilidad actual de la API de gestión para cargar, descargar, inspeccionar, configurar y ejecutar acciones (p. ej. crear e iniciar nuevas instancias). Cada proveedor tiene su propia API para evitar que el software de orquestación funcione con distintos proveedores de servicios. Existen varios enfoques para este problema. Algunos colectivos como [Open Grid Forum](#) **intentan crear el estándar** conocido como **OCCI (Open Cloud Computing**

[Interface](#)). Otros como [Eucalyptus](#) emulan la interfaz de los [servicios Web de Amazon](#) como estándar válido. VMware ha desarrollado su propia [API vCloud](#), la cual [envió a la DMTF \(Distributed Management Task Force\) como estándar abierto](#). La API vCloud ofrecerá una base de interoperabilidad entre los [proveedores de servicios basados en VMware](#) (y posiblemente otros proveedores en el futuro), pero [casi con total seguridad no a los jugadores establecidos](#). La mayoría de los proveedores renuncian a la estandarización oficial porque quieren (y necesitan) moverse rápidamente en este mercado en constante evolución. Además, los organismos normalizadores no son famosos por su rapidez precisamente. Sin embargo, el hecho de que no se adopte una API única para todo el sector no tiene por qué impedir la portabilidad e interoperabilidad.

Es posible combinar varias [APIs](#) en una sola API, incluso sin la participación de los proveedores. En el espacio de la virtualización, el paquete [libvirt](#) ofrece una API para las APIs. Asimismo, en el ámbito de la computación en nube, un grupo de expertos ya ha asumido esta tarea mediante el proyecto [UCI \(Unified Cloud Interface Project\)](#), aunque este se encuentra aún en su fase inicial.

Otra iniciativa, [cloudloop](#), incluye una API para trabajar con múltiples servicios de almacenamiento. Estos tipos de APIs diseñadas para múltiples APIs [facilitan un modo de interoperabilidad, mediante el que los proveedores de estructuras y middleware y usuarios finales pueden consumir una única API sin preocuparse por depender de un solo proveedor de servicios](#).

En cuanto al concepto de plataforma como servicio (PaaS), la portabilidad e interoperabilidad constituyen un desafío aún mayor. Los formatos de los datos para los servicios de plataforma suelen ser completamente diferentes. Así por ejemplo, [Windows Azure](#) suministra [servicios de bases de datos y contenedores de aplicaciones .NET](#). Las aplicaciones y los datos de Azure no son compatibles con [Google AppEngine](#) y viceversa. La única forma de evitar la dependencia de un único proveedor cuando se utiliza PaaS es elegir una estructura facilitada por varios proveedores y evitar extensiones específicas de un proveedor (como las extensiones Python de AppEngine).




Probablemente veremos estrategias de abstracción parecidas donde se puedan desarrollar aplicaciones ejecutables en muchas soluciones PaaS. Imagino que se producirá un gran desarrollo en este ámbito a medida que la carga de trabajo pasa de IaaS a PaaS.

El software como servicio (SaaS) supone el reto principal debido a la inherente diversidad de datos. Al igual que sabemos que los datos de Facebook no se pueden exportar ni importar de otra red social (Matt Asay lo denominó [Hotel California of Tech](#)), tampoco debemos suponer que todos los servicios de software ofrecen extracción de datos. Solo se acepta este hecho cuando el servicio ofrecido no tiene estándares. En casos como el de [Google Docs](#), es lógico esperar algún tipo de conversión como las [nuevas opciones de exportación de Google lanzadas esta semana](#). En este entorno, la conversión es una vía mucho más práctica para la portabilidad que la estandarización.

En el mercado de la computación en nube de rápida evolución, podemos esperar que [surjan de múltiples estándares](#) pero, como Stephen Foskett afirma, [“solo sobrevivirán los estándares útiles”](#). Se trata de un proceso saludable en un entorno nuevo. Mientras tanto, es posible lograr portabilidad e interoperabilidad independientemente de los estándares. Podemos acabar con el problema de depender de un solo proveedor y garantizar la disponibilidad de los servicios y datos mediante conversión y abstracción.

Esta entrada se realizó el Wednesday, – 28. October 2009 y se encuentra en la categoría [Cloud Computing](#), [Seguridad & Internet](#). Seguimiento de los comentarios de esta entrada mediante [RSS 2.0](#) Feed. Puede escribir [un comentario](#), o introducir un [vínculo de referencia](#) desde su sitio.

## Leyenda

Pregunta	Color
Identificar los estándares que se mencionan	
Identificar ventajas de cada estándar	
Identificar los modos de desarrollo e implantación de los estándares	

## **Actividades**

### → ¿Ideas principales?

La idea principal es informarnos sobre los estándares de computación en la nube. Nos expone las ventajas, competencia y desarrollo de estandarización aplicándolos a la computación en la nube. Finaliza con una idea de cómo va a evolucionar la computación en la nube en un futuro próximo.

### → ¿Orden del texto?

Podemos encontrar varios órdenes

- ◆ Hardware → Software → Usuario
- ◆ Pasado → Presente → Futuro
- ◆ Introducción → Desarrollo → Conclusión
- ◆ Problema → Solución