

## INVESTIGACIÓN

## Acceso abierto

## Orquestación de recursos en la nube en el panorama de múltiples nubes: una revisión sistemática de los marcos existentes

Orazio Tomarchio <sup>†</sup> \* , Domenico Calcaterra <sup>†</sup> y Giuseppe Di Modica <sup>†</sup>

## Resumen

El número tanto de proveedores de servicios que operan en el mercado de la nube como de clientes que consumen servicios basados en la nube aumenta constantemente, lo que demuestra que el paradigma de la computación en la nube ha desarrollado con éxito su potencial. Sin embargo, el crecimiento incesante del mercado de la nube plantea duros desafíos a sus participantes. Del lado del proveedor, la capacidad de orquestar los recursos para maximizar las ganancias sin fallar las expectativas de los clientes es motivo de preocupación. Del lado del cliente, la selección eficiente de recursos de una pléthora de servicios similares anunciados por una multitud de proveedores es una pregunta abierta. En este panorama de múltiples nubes, varias iniciativas de investigación abogan por el empleo de marcos de software (es decir, marcos de orquestación de recursos en la nube - CROF) capaces de orquestar los recursos heterogéneos ofrecidos por una multitud de proveedores de la nube de la manera que mejor se adapte a las necesidades del cliente. El objetivo de este documento es proporcionar al lector una revisión sistemática y una comparación de los CROF más relevantes que se encuentran en la literatura, así como resaltar los problemas abiertos de la computación en múltiples nubes que la comunidad de investigación debe abordar en el futuro próximo.

**Palabras clave:** computación en la nube, orquestación de recursos en la nube, nubes múltiples, interoperabilidad de la nube, nubes interconectadas, intermediación en la nube

## Introducción

En los últimos años, la computación en la nube se ha consolidado como un nuevo modelo de computación distribuida al ofrecer servicios complejos de hardware y software en muy diferentes campos. Como se informó en el Informe sobre el estado de la nube de RightScale 2019 [1], muchas empresas y organizaciones han adoptado con éxito el paradigma de la computación en la nube en todo el mundo, mientras que cada vez más se acercan a él porque ven una oportunidad real de hacer crecer su negocio. Según ese informe, el 94 % de los profesionales de TI encuestados dijeron que sus empresas usan servicios de computación en la nube y el 91 % usa la nube pública. Las organizaciones aprovechan casi 5 nubes en promedio y las empresas están

ejecutando alrededor del 40 por ciento de sus cargas de trabajo en la nube. El gasto en la nube empresarial está creciendo rápidamente, ya que las empresas planean gastar un 24 % más en la nube pública en 2019 en comparación con 2018.

La competencia entre los proveedores de la nube es cada vez más fuerte para adquirir cuotas de mercado cada vez mayores: un punto clave para optimizar el uso de los recursos y explotar al máximo el potencial de la computación en la nube es el tema de la orquestación de recursos [2]. La orquestación de recursos en la nube se refiere a operaciones complejas como selección, implementación, monitoreo y control de recursos en tiempo de ejecución. El objetivo general de la orquestación es garantizar la entrega completa y sin problemas de las aplicaciones mediante el cumplimiento de los objetivos de calidad de servicio (QoS) tanto de los propietarios de aplicaciones en la nube como de los proveedores de recursos en la nube. La orquestación de recursos se considera una actividad desafiante debido a la dimensión de escala que han alcanzado los recursos y la proliferación

\* Correspondencia: [orazio.tomarchio@unict.it](mailto:orazio.tomarchio@unict.it)<sup>†</sup> Orazio Tomarchio, Domenico Calcaterra y Giuseppe Di Modica contribuyeron igualmente a este trabajo.

Departamento de Ingeniería Eléctrica, Electrónica e Informática, Universidad de Catania, Catania, Italia

de proveedores de nube heterogéneos que ofrecen recursos a diferentes niveles de la pila de nubes.

Marcos de orquestación de recursos en la nube (CROF) han surgido como sistemas para gestionar el ciclo de vida de los recursos, desde la fase de selección hasta la de seguimiento [2–4]. Hoy en día, la mayoría de los proveedores comerciales de nube ofrecen una plataforma de orquestación en la nube para los usuarios finales [5]: sin embargo, estos productos son patentados y, por razones comerciales obvias, no son portátiles. Además, aunque moderno existen soluciones de gestión de configuración (p. ej., Amazon OpsWorks, Ansible, Puppet, Chef) que brindan soporte para manejar la configuración de recursos a través de servicios en la nube, todos los usuarios potenciales (desde programadores profesionales y administradores de sistemas hasta usuarios finales no expertos) a menudo se les exige que entiendan varias nubes de bajo nivel API de servicio y construcciones de programación de procedimientos en para crear y mantener configuraciones de recursos complejas.

El advenimiento de la computación en múltiples nubes exagera aún más los ya desafiantes problemas de orquestación. El paradigma multinube es una tendencia tecnológica muy reciente dentro del panorama de la computación en la nube, que gira en torno a la oportunidad de aprovechar los servicios y recursos proporcionados por múltiples nubes [6, 7]. La multinube supone que no existe un acuerdo a priori entre proveedores de la nube, y un tercero es responsable de la servicios. Ese es el caso de los escenarios de intermediación en la nube, donde un corredor actúa como intermediario entre los proveedores de la nube y consumidores de la nube [8]. Para permitir un paradigma multinube eficaz, es esencial garantizar una fácil portabilidad de las aplicaciones entre los proveedores de la nube [9, 10]. Este nuevo requisito exige más potentes mecanismos de orquestación de recursos transversales a múltiples dominios administrativos de la nube, es decir, capaces de tratar con la heterogeneidad de los recursos de nube subyacentes y servicios.

Este trabajo explora los muchos problemas de la organización de recursos en el entorno de la nube. Una revisión de las obras existentes en el campo direccionado se lleva a cabo con el fin de identificar el desafíos que han atraído principalmente a los investigadores en los últimos años, y resaltar los aspectos que no han sido plenamente cubierto todavía. La principal contribución de nuestro trabajo es doble. En primer lugar, analizando en profundidad la literatura de aparición reciente, construimos una taxonomía completa de características deseables y dimensiones útiles para caracterizar los CROF. Entonces, en De acuerdo con las características identificadas, comparamos varios CROF tanto de la industria como de la academia. Esta voluntad ayudar al lector no sólo a entender las fortalezas de cada marco, sino también para identificar los desafíos no resueltos que deben abordarse en el futuro cercano.

El resto del documento está organizado de la siguiente manera. En el apartado "Metodología de la investigación" se describe la metodología seguida en nuestro estudio. Sección "Encuestas relacionadas" presenta un relevamiento de trabajos existentes relacionados con nuestro estudio.

En la sección "Marco de análisis" identificamos el CROF capacidades que se han utilizado para llevar a cabo la revisión presentado en la sección "Revisión de los cROF". En "Enfermedad crítica" sección "cusión" resumimos los resultados de la revisión, enfatizando las limitaciones actuales y los desafíos abiertos. Finalmente, la sección "Conclusión" concluye nuestro trabajo.

## Metodología de investigación

La principal motivación de este estudio es arrojar luz sobre los recientes avances que tanto la industria como la academia han hecho para enfrentar los problemas de orquestación de recursos en la nube en el paisaje de múltiples nubes.

Con este objetivo en mente, identificamos los campos relevantes para nuestro estudio con el fin de enmarcar claramente la investigación. alcance. Más allá del esperado tema de organización de recursos en la nube, también se investigaron los siguientes temas macro: interoperabilidad de la nube, intermediación en la nube, nubes interconectadas. Como se describe en la sección "Introducción", la nube

La orquestación de recursos se ocupa del descubrimiento, selección, asignación y gestión de los recursos de la nube. Cuando existen múltiples nubes, los problemas de correa e interoperabilidad de la nube debido al acceso simultáneo a servicios heterogéneos de proveedores interconectados no pueden solucionarse. descuidado en el análisis de la orquestación de recursos en la nube.

Examinamos la literatura producida recientemente en el campos mencionados. Específicamente, buscamos propuestas, marcos, prototipos, productos comerciales de alguna manera abordando los temas discutidos anteriormente. las bases de datos tomados en consideración en esta encuesta son los siguientes: Scopus1, ACM Digital Library2, IEEE Xplore Digital Library3, Elsevier ScienceDirect4 y SpringerLink5. Nosotros también se ocupó de filtrar los elementos de investigación que están fechados anterior a la última década.

Descubrimos que muchos investigadores ya han publicado encuestas que son relevantes para nuestro objeto de estudio. Cada una de estas encuestas enumera y clasifica, bajo diferentes perspectivas, numerosas iniciativas realizadas bajo el gran paraguas del campo de orquestación de recursos en la nube, sea ellos CROF completos o propuestas menores que se enfocan solo en un conjunto restringido de funciones de orquestación. El objetivo principal del estudio propuesto en este trabajo es proporcionar un análisis nuevo y unificado de las iniciativas existentes, que abarca todas las perspectivas de análisis propuestas por las encuestas pasadas y finalmente identifica la falta unos.

Por lo tanto, como se muestra en la Fig. 1, el primer paso de nuestro estudio consistió en revisar los levantamientos bibliográficos con el objetivo de a) consolidar la lista de CROF y, en general, las propuestas sobre las que realizar un análisis comparativo cualitativo,

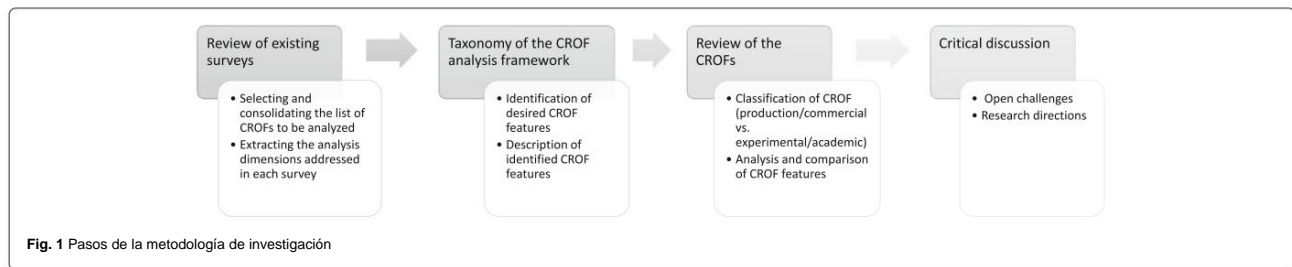
1<http://www.scopus.com/>

2<https://dl.acm.org/>

3<http://ieeexplore.ieee.org/>

4<https://www.sciencedirect.com/>

5<https://link.springer.com/>



b) extraer las dimensiones de análisis abordadas en cada encuesta. Luego, el segundo paso fue elaborar un marco de análisis para proporcionar una visión más completa conjunto de características en las que un nuevo paso de comparación ser ejecutado A continuación, siguiendo las referencias encontradas en las encuestas, cada CROF de la lista se revisó más de acuerdo con a las directrices comparativas antes mencionadas, y la El resultado del análisis finalmente se reunió en una tabla sinóptica, lo que ayudó al lector a comparar los diferentes características. Finalmente, los resultados de la comparación fueron la base para una discusión crítica sobre el estado del arte, desafíos abiertos y expectativas futuras sobre los CROF.

### Encuestas relacionadas

Esta sección presenta los resultados de una revisión de la literatura que llevamos a cabo para identificar los estudios publicados que se relacionan con nuestro trabajo en diversos grados. Específicamente, nosotros investigó la vasta área de búsqueda de computación en la nube para propuestas e iniciativas pertenecientes a la nube temática orquestación de recursos en el panorama de múltiples nubes.

De particular importancia en el contexto de la discusión fueron los siguientes trabajos: Inter-cloud Challenges, Documento de posición del Grupo Sectorial de Expectativas y Problemas [11], y el Manifiesto para la Computación en la Nube de la Generación Futura [12]. Ambos trabajos reconocieron el aprovisionamiento de recursos y la orquestación como un desafío abierto. En [11], Ferrer et al. la reconoció como un área de investigación con un alto impacto empresarial a medio plazo. Además, a la luz de recursos de nube cada vez más heterogéneos distribuidos en diversas tipologías y modelos de nube, tanto Los estudios destacaron la importancia de investigar áreas de investigación, como la interoperabilidad y la portabilidad de la nube, el descubrimiento y la composición de servicios (es decir, el corretaje de la nube) y las nubes interconectadas. la relacion de estas áreas de investigación relacionadas con el tema principal de este encuesta se representan esquemáticamente en la Fig. 2. Representamos el alcance de la investigación de orquestación de recursos de nubes múltiples como un gran paraguas que cubre completamente el área de investigación de orquestación de recursos en la nube y comparte parcialmente los temas cubiertos por los campos de investigación de corretaje en la nube, internube e interoperabilidad/portabilidad de la nube.

Hacemos notar que el estudio realizado en este primer paso investigativo no pretendía buscar propuestas reales e iniciativas en los campos enfocados. En cambio, apuntó

las obras literarias que se proponen sondeos de la propuestas más relevantes (paso 1 en la Fig. 1). Aquí, el objetivo es resaltar los límites de las encuestas bibliográficas existentes y, así, proporcionar una motivación a nuestro trabajo. Además, al "evaluar las encuestas bibliográficas existentes" pudimos recopilar los punteros a las propuestas de investigación reales, que fueron objeto de investigación en los siguientes pasos de nuestro estudio.

A continuación, comentamos algunas de las encuestas bibliográficas más representativas desglosadas en las cuatro mencionadas anteriormente. subtemas de la nube. En cada una de las siguientes secciones se introduce brevemente el subtema y los aspectos relevantes para el se señala el tema de orquestación de nubes múltiples.

### Interoperabilidad en la nube

La comunidad de computación en la nube generalmente usa el término interoperabilidad para referirse a la capacidad de moverse fácilmente cargas de trabajo y datos de un proveedor de nube a otro o entre nubes públicas y privadas [13]. Hace diez años, el organismos de estandarización NIST [14], OMG [15] y DMTF [16] desarrolló, entre otros, varios casos de uso relacionados a la interoperabilidad en la nube. Todos los órganos, independientemente de entre sí, definieron un paraguas común de casos de uso de interoperabilidad que abarcan temas como la autenticación de usuarios, migración de cargas de trabajo, migración de datos y gestión de cargas de trabajo.

En [17], los autores realizaron una encuesta exhaustiva en la interoperabilidad de la nube, con un enfoque en la interoperabilidad entre diferentes plataformas de nube IaaS. investigaron los esfuerzos existentes sobre taxonomías y estandarización de interoperabilidad de la nube e identificó algunos problemas abiertos para avanzar en el tema de investigación también. Sin embargo, las soluciones y conceptos presentados se centran principalmente en IaaS interoperabilidad

En [18], los autores realizaron su encuesta sobre la interoperabilidad y portabilidad de servicios en sistemas en la nube con respecto a descubrimiento de servicios de computación en la nube. Aún así, otros enfoques de interoperabilidad como la Ingeniería Dirigida por Modelos (MDE) y las soluciones abiertas no fueron ampliamente exploradas.

En [19], los autores describen los principales desafíos sobre la federación de la nube y la interoperabilidad, así como mostró y revisó los estándares potenciales para abordar estos asuntos. Similar a [17], su trabajo está restringido a Interoperabilidad IaaS, sin ningún otro servicio o implementación modelos que se cubren.

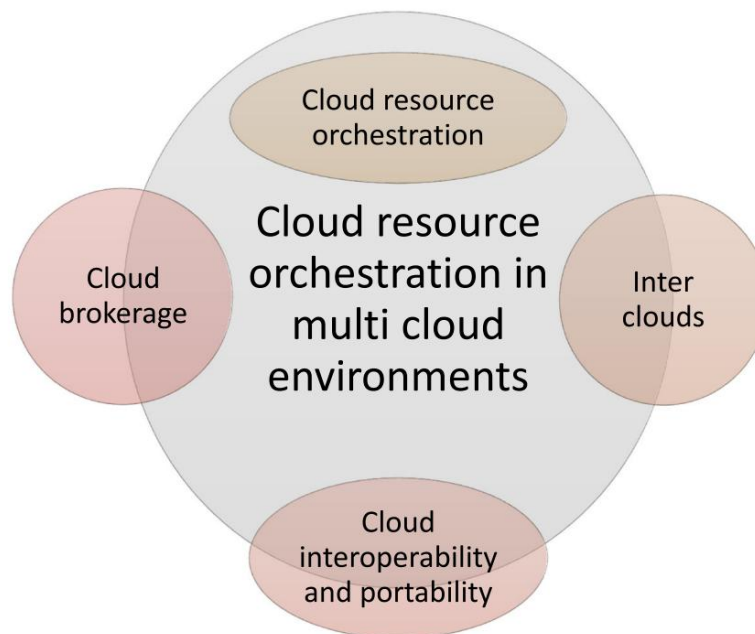


Fig. 2 Áreas de investigación relacionadas

### Corretaje en la

**nube** De acuerdo con la definición de Gartner [20], “el corretaje de servicios en la nube es una función de TI y un modelo comercial en el que una empresa u otra entidad agrega valor a uno o más servicios en la nube (públicos o privados) en nombre de uno o más consumidores. de ese servicio a través de tres roles principales que incluyen agregación, integración y corretaje de personalización”. Según lo define NIST [21], un corredor de servicios en la nube “... es una entidad que administra el uso, el rendimiento y la entrega de servicios en la nube y negocia las relaciones entre los proveedores de la nube y los consumidores de la nube”. A partir de estas definiciones, queda claro que cualquier actor empresarial que pretenda actuar como intermediario entre los consumidores de la nube y los proveedores de la nube debe hacer frente a la diversidad de proveedores y la heterogeneidad de la multitud de servicios que ofrecen estos últimos.

En [6], los autores propusieron taxonomías para arquitecturas entre nubes y intermediación de aplicaciones. Presentaron un estudio detallado de los desarrollos académicos e industriales para la internube, catalogando muchos proyectos y ajustándolos a las taxonomías presentadas. También analizaron los trabajos existentes e identificaron desafíos abiertos en el área de intermediación de aplicaciones entre nubes. Sin embargo, sus esfuerzos se limitan a las estrategias basadas en corredores.

En [22], se realizó una encuesta sistemática de literatura para recopilar estudios relacionados con el corretaje en la nube. Los autores presentaron una comprensión del estado del arte y una taxonomía novedosa para caracterizar los intermediarios en la nube, identificando las principales limitaciones de las soluciones actuales y destacando áreas para futuras investigaciones. Sin embargo, al igual que [6], todo su análisis solo cubre los enfoques basados en intermediarios.

### Nubes interconectadas

Las nubes interconectadas, también llamadas internubes, pueden verse como una evolución natural de la computación en la nube. Inter cloud ha sido presentado por Cisco [23] como una “nube de nubes” global interconectada que imita el término Internet, “red de redes”. Básicamente, Inter-cloud se refiere a una malla de nubes que se unifican en base a protocolos estándar abiertos para proporcionar una interoperabilidad en la nube.

El Global Inter-cloud Technology Forum (GICTF) ofrece una definición más sofisticada de Inter-cloud.

[24]: “Inter-nube es un modelo de nube que, con el fin de garantizar la calidad del servicio, como el rendimiento y la disponibilidad de cada servicio, permite la reasignación de recursos bajo demanda y la transferencia de carga de trabajo a través de un interfuncionamiento de sistemas en la nube de diferentes proveedores de nube basados en la coordinación de los requisitos de calidad de servicio de cada consumidor con el SLA de cada proveedor y el uso de interfaces estándar”.

En [8, 9, 25], el autor investigó el consumo de recursos y servicios de múltiples nubes, además de proponer una lista de requisitos para soluciones de interoperabilidad, destacando las barreras tecnológicas y algunas soluciones conocidas para multi-nube. ambientes. La autora no presentó el origen de estos requisitos, ni identificó el grado de cumplimiento de los requisitos por enfoques teóricos y soluciones técnicas.

En [26], los autores discutieron todos los aspectos relevantes que motivan la interoperabilidad de la nube, categorizando e identificando escenarios y arquitecturas de interoperabilidad de la nube. Proporcionaron una taxonomía de los principales desafíos para el

Realización entre nubes. Una revisión exhaustiva de la También se realizó el estado del arte, incluyendo iniciativas de estandarización, proyectos en curso y estudios en el área.

En [27], los autores analizaron la literatura existente para identificar cómo ha sido la interoperabilidad en la computación en la nube dirigido. Investigaron los requisitos y los escenarios de uso de las aplicaciones interoperables, así como de la nube. soluciones de interoperabilidad, presentando una lista limitada de soluciones abiertas problemas y direcciones para futuras investigaciones.

En [28], los autores examinaron la literatura para analizar y categorizar varias soluciones para resolver los problemas de interoperabilidad y portabilidad de las nubes interconectadas, refiriéndose tanto al lado del usuario (múltiples nubes o agregado servicio por Broker) y del lado del proveedor (nubes federadas o Nubes híbridas), como se especifica en [8, 25]. Ellos también realizó un análisis comparativo de las obras de la literatura en la misma categoría, y discutió los desafíos de las nubes interconectadas en la misma línea que [17] y [26].

A pesar de profundizar en las nubes interconectadas, a partir de con motivación, escenarios, posibles soluciones para la interoperabilidad y terminando con problemas abiertos y direcciones futuras, todos estos trabajos ([26–28]) prestaron una atención limitada a orquestación de recursos en la nube. Además, ninguno de ellos cubrió aspectos relacionados con el desarrollo de aplicaciones, implementación y gestión del ciclo de vida.

### Orquestación de recursos en la nube

En un panorama donde las organizaciones llegan a utilizar muchos tipos de sistemas de computación en la nube simultáneamente, la complejidad de las cargas de trabajo dedicadas a la gestión de el ciclo de vida de los recursos (datos y aplicaciones) a través los sistemas aumentan dramáticamente. Orquestación en la nube es el proceso de administrar estas múltiples cargas de trabajo, de forma automatizada, a través de varias soluciones en la nube. Las actividades típicas que subyacen a un proceso tan complejo son la descripción, selección, configuración, despliegue, seguimiento y control de los recursos. No olvidemos que el El problema de orquestación se ve exacerbado por la diversidad de los sistemas en la nube, por lo que se refiere tanto a aspectos técnicos como características administrativas.

En [2], los autores caracterizaron el recurso de la nube orquestación en una pila de varias capas, y resaltado los principales desafíos de investigación involucrados en la programación operaciones de orquestación para diferentes recursos en la nube tipos en todas las capas de una pila de recursos en la nube. el alcance de su análisis se restringe, sin embargo, al área de orquestación de recursos en la nube.

En [3], los autores propusieron una taxonomía multidimensional para clasificar y comparar técnicas de gestión de recursos en la nube tanto de la industria como de la academia. identificar temas de investigación abiertos y ofrecer direcciones para estudio futuro. Similar a [2], su trabajo solo cubre el tema de orquestación de recursos en la nube.

En [29], los autores realizaron una literatura sistemática encuesta para construir una taxonomía de los principales intereses de investigación sobre TOSCA. Se abordaron diferentes temas, tales como el diseño de métodos de orquestación en la nube usando TOSCA, que amplía el lenguaje de TOSCA y presenta herramientas para manipular modelos TOSCA. A pesar de ser Concebido como un tema que se espera que desempeñe un papel cada vez más importante, la interoperabilidad recibió una recepción muy limitada. atención.

## Marco de análisis

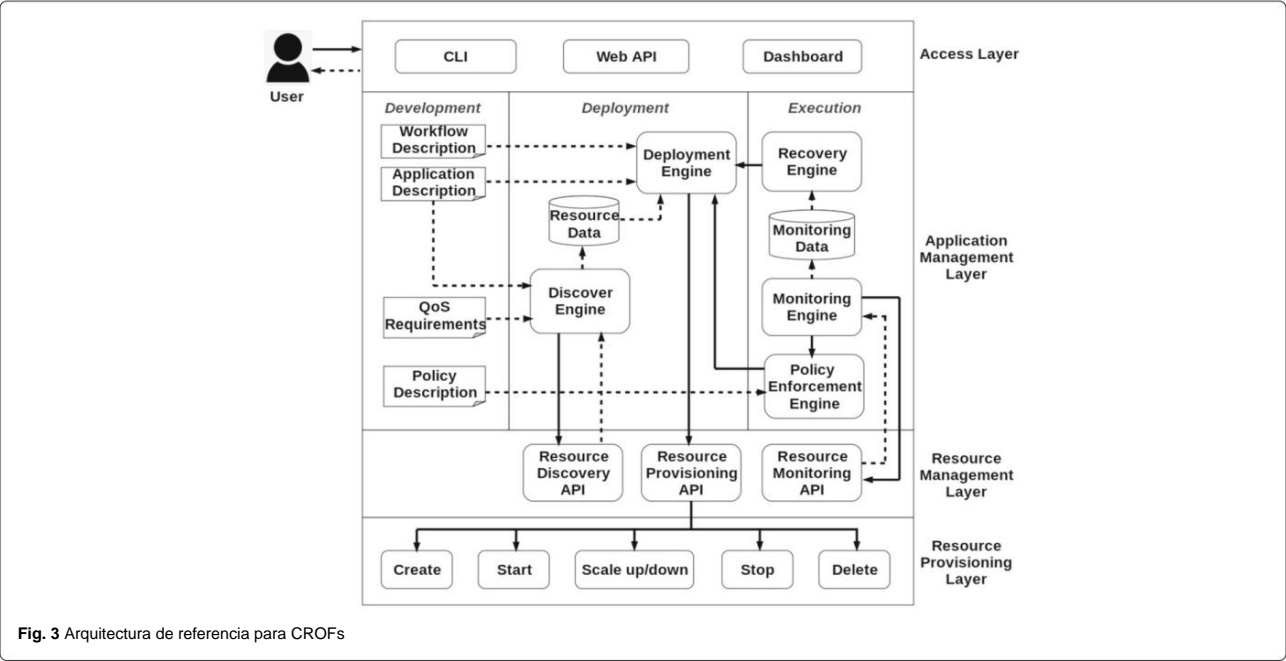
En esta sección presentamos las capacidades deseadas para CROF, centrándose en el despliegue y la gestión aspectos. Desde el punto de vista de los consumidores, los CROF implementan un modelo orientado a servicios que garantiza el alojamiento y la entrega exitosos de aplicaciones mediante el uso de la nube. recursos para cumplir con sus requisitos de QoS. Nuestro la arquitectura de referencia para CROF se muestra en la Fig. 3. Los procesos y servicios involucrados en la organización de recursos en la nube se clasifican según sus funcionalidades. en relación con este modelo de referencia.

La capa de acceso regula la interacción con el marco de trabajo. Los usuarios pueden acceder a los servicios de las capas inferiores mediante CLI, API web y paneles. La capa de gestión de aplicaciones se refiere al manejo de las aplicaciones a lo largo de todo su ciclo de vida, desde el desarrollo hasta la ejecución, pasando por el despliegue. El Desarrollo se refiere a lenguajes y modelos para

normalmente representan aplicaciones, flujos de trabajo, requisitos de QoS y políticas. Las descripciones de la aplicación definen los componentes de la aplicación, así como sus relaciones. Las descripciones del flujo de trabajo especifican los aspectos de comportamiento de las aplicaciones por medio de enfoques declarativos o imperativos.

Las descripciones de las políticas brindan a las aplicaciones dinámicas comportamientos de control (por ejemplo, la definición de políticas basadas en la carga para escalar hacia arriba y hacia abajo las aplicaciones) para cumplir con QoS requisitos La implementación se refiere a la implementación real de la aplicación en los recursos de la nube, que podría ir a través de un proceso preliminar de descubrimiento de recursos. los La ejecución implica la automatización efectiva de tareas de gestión complejas, como el escalado y el manejo de fallas, que generalmente requieren un sistema de recolección de motor de monitoreo y métricas de aplicación. Según las métricas capturadas, un motor de recuperación y un motor de cumplimiento de políticas pueden determinar las decisiones a tomar para recuperarse de fallas y hacer cumplir las políticas, respectivamente.

La capa de gestión de recursos incluye servicios (por ejemplo, servicios de descubrimiento, servicios de aprovisionamiento, servicios de monitoreo) que manejan recursos a lo largo de todo su ciclo de vida. Estos servicios coordinan las acciones requeridas desde el capa superior aprovechando las operaciones en la capa de aprovisionamiento de recursos. La capa de aprovisionamiento de recursos abarca los servicios que ofrecen las operaciones más básicas con respecto a los recursos de la nube. Una gama de servicios de aprovisionamiento (por ejemplo,



crear, iniciar, escalar, detener y eliminar) generalmente se proporcionan para cada recurso admitido.

Se han realizado importantes investigaciones en el ámbito académico y de la industria hacia la caracterización de las herramientas de orquestación de la nube. En [4, 30], Baur et al. investigó las características requeridas para tales herramientas y dio una definición de ellas. En [31], Ranjan et al. introdujo dimensiones técnicas para el análisis CROF, proporcionando así información sobre los marcos existentes. En [3], Weerasiri et al. identificó las dimensiones principales y los componentes básicos comunes que caracterizan las soluciones de orquestación de recursos en la nube.

En [32–34], los autores presentaron su visión de la computación en la nube, incluidos puntos de vista sobre futuras áreas de investigación, una de las cuales es el aprovisionamiento y la orquestación de recursos. Se llevó a cabo un análisis exhaustivo de estas áreas de investigación y los desafíos relacionados desde diferentes perspectivas.

En [35], GigaSpaces Research investigó enfoques predominantes para administrar aplicaciones en entornos de nube, a saber, orquestación, PaaS (plataforma como servicio) y CMP (plataforma de administración en la nube). Se propusieron varias categorías que sirvieron como base común para la comparación entre los diferentes enfoques.

Con base en el estudio de Baur et al.[4], enriquecimos la lista de capacidades deseables pertenecientes a los CROF al revisar la literatura e integrar los trabajos mencionados anteriormente. Dichas capacidades, resumidas en la Fig. 4, se pueden clasificar en dos categorías principales como Funciones de la nube o Funciones de la aplicación. Los detalles sobre cada conjunto de funciones se proporcionan en las siguientes subsecciones.

**Funciones de la nube**

Las funciones de la nube abordan los aspectos de la infraestructura de la nube con un enfoque especial en la implementación compatible en varios proveedores de la nube. Mientras que algunos trabajos [4, 35] investigaron características como la compatibilidad con nubes múltiples o entre nubes y la integración de servicios y sistemas externos, otros [31] se centraron en capacidades como la interoperabilidad y los modos de acceso a CROF. Proponemos un enfoque integral que tenga en cuenta todos los aspectos mencionados que discutimos a continuación.

**Compatibilidad con múltiples nubes** La compatibilidad con múltiples proveedores de nube es una de las características más cruciales para los CROF, ya que permite seleccionar la mejor oferta para una aplicación de un panorama de nube diverso. Los proveedores de la nube a menudo difieren entre sí con respecto a sus API. Por esa razón, los CROF deben ofrecer una capa de abstracción en la nube (consulte la sección "Enfoque de interoperabilidad" ), que oculta las diferencias y evita la necesidad de una personalización específica del proveedor que cause el problema de bloqueo del proveedor.

**Compatibilidad entre nubes** La compatibilidad entre nubes mejora la característica de múltiples nubes al permitir distribuir instancias de componentes de una sola aplicación entre múltiples proveedores de nube. Las ventajas de la implementación entre nubes son tres: a) permite una selección sofisticada de los proveedores de nube que mejor se adaptan por instancia de componente, optimizando costos o mejorando la calidad de los servicios; b) aprovecha la disponibilidad de la aplicación ya que introduce resiliencia.



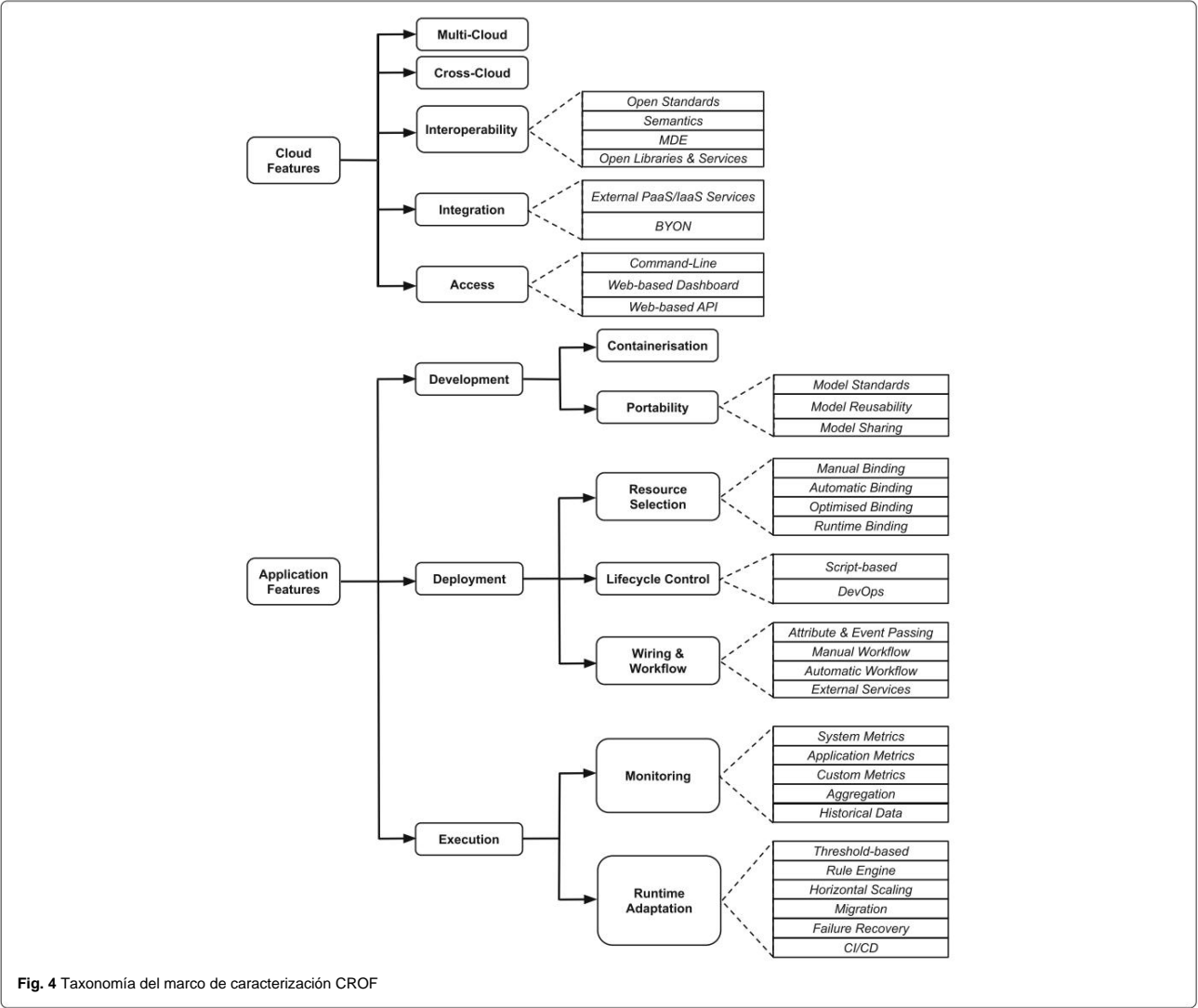


Fig. 4 Taxonomía del marco de caracterización CROF

falla de proveedores de nube individuales; yc) ayuda a hacer frente a los problemas de privacidad.

Enfoque de interoperabilidad

En el contexto de la computación en la nube, la interoperabilidad se puede definir como la capacidad de desarrollar aplicaciones que combinan recursos que pueden interoperar o trabajar juntos desde múltiples proveedores de la nube, aprovechando así las características específicas proporcionadas por cada proveedor [27]. Algunos trabajos de investigación [9, 27, 28] revisaron exhaustivamente la literatura para diseccionar el estado del arte en la interoperabilidad de la nube, lo que resultó en una amplia gama de enfoques que se clasifican en las siguientes categorías: estándares abiertos, semántica, ingeniería basada en modelos (MDE) y bibliotecas y servicios abiertos.

La formulación de estándares para la computación en la nube es la solución más obvia para la interoperabilidad. Aunque hasta ahora se han propuesto una plétora de normas (p. ej.,

OCCI6, CIMI7, OVF8, CDMI9, TOSCA [36]), la falta de estándares ampliamente aceptados requiere investigar otras soluciones para la interoperabilidad. Cuando los proveedores de la nube usan diferentes API y modelos de datos para exhibir las mismas características, la interoperabilidad semántica se vuelve involucrada. Las tecnologías semánticas (p. ej., OWL10, SPARQL11, SWRL12) pueden resultar útiles para proporcionar interoperabilidad semántica entre diferentes proveedores de nube. Los enfoques basados en intermediarios también pueden aliviar la interoperabilidad semántica por medio de interfaces basadas en ontologías que ocultan las diferencias entre los proveedores de la nube. La interoperabilidad de la nube también se puede abordar mediante la explotación de técnicas MDE [10].

6<https://occi-wg.org/about/specification/>  
7<https://www.dmtf.org/standards/cmwwg> 8<https://www.dmtf.org/standards/ovf> 9<https://www.snia.org/cdmi> 10<https://www.w3.org/TR/owl-syntax/> 11<https://www.w3.org/TR/sparql11-query/> 12<https://www.w3.org/Submission/SWRL/>

Otra solución viable para la interoperabilidad de la nube incluye bibliotecas abiertas (p. ej., Apache jclouds<sup>13</sup>, Apache Libcloud<sup>14</sup>) y servicios, que se basan en capas de abstracción para desvincular el desarrollo de aplicaciones del propietario tecnologías de los proveedores de la nube.

## Integración

Soporte para servicios IaaS/PaaS avanzados (p. ej., DBaaS, LBaaS, FWaaS) es deseable. Reduce la complejidad y esfuerzos de gestión para el usuario final. En una nota negativa, viene a expensas de la flexibilidad.

BYON (Bring Your Own Node) captura la capacidad de usar servidores que ya están en ejecución para la implementación de aplicaciones. En particular, permite el uso de servidores no gestionados por una plataforma en la nube o máquinas virtuales en una nube no compatible proveedores

## Acceso

Esta característica captura qué interfaces utilizan los CROF para interactuar con los recursos de la nube. Tres tipos de interfaces son generalmente compatible: línea de comandos, panel de control basado en la web, y API basada en web.

Las interfaces de línea de comandos envuelven la API específica de la nube acciones como comandos o scripts ejecutables a través de shell entornos. A pesar de que las interfaces de línea de comandos son más fácil de implementar, su uso requiere una comprensión profunda de los recursos de la nube y la orquestación relacionada operaciones.

Los paneles basados en la web presentan los recursos de la nube como artefactos y catálogos de recursos fáciles de usar. Los artefactos visuales y los catálogos tienen como objetivo simplificar la selección, el ensamblaje y la implementación de recursos. Estas características hacen Tableros basados en web más simples y más flexibles que interfaces de línea de comandos.

Las API basadas en web permiten otras herramientas y sistemas (p. ej. herramientas de monitorización) para integrar las operaciones de gestión de recursos en la nube en sus funcionalidades. Ellos proveen la mayor abstracción de los tres tipos de interfaz.

## Características de la aplicación

Las características de la aplicación abordan el desarrollo, la implementación, y aspectos de ejecución de las aplicaciones. Para ello, a diferencia de todos los trabajos anteriores, recopilamos características de acuerdo con el fase de solicitud a la que pertenecen. Por ejemplo, con referencia a la fase de desarrollo, hemos identificado

Portabilidad y Containerización como características relevantes. Además, también proponemos una clasificación de aplicación dominios de interés para CROFs.

## Dominio de aplicación

El dominio de aplicación se refiere a los tipos de aplicaciones que Los CROF se han diseñado y personalizado para. Académico

Se han realizado investigaciones para la caracterización de

dominios de aplicación en los últimos años [31][37][38].

Basándose en el estudio de Buyya et al. [37], clasificamos los dominios de aplicación en dos categorías: Científico aplicaciones y aplicaciones empresariales (ver Fig. 5).

Los sistemas de computación en la nube satisfacen las necesidades de diferentes tipos de aplicaciones en el dominio científico: aplicaciones informáticas de alto rendimiento (HPC), aplicaciones informáticas de alto rendimiento (HTC) y análisis de datos a gran escala/Internet de las cosas (IoT), que es un asunto de interés común tanto para el sector científico como para el empresarial. En lo que respecta al negocio dominio, la computación en la nube es la tecnología preferida para una amplia gama de aplicaciones, desde aplicaciones web de múltiples niveles (p. ej., aplicaciones web, móviles, de juegos en línea) hasta redes de entrega de contenido y medios (CDN) aplicaciones (por ejemplo, codificación y transcodificación de video, reproducción de video, transmisión de video, contenido web/móvil aceleración).

## Portabilidad

La portabilidad se ha definido como la capacidad de un programa para ejecutarse en varios tipos de procesamiento de datos. sistemas sin convertir el programa a otro lenguaje y con poca o ninguna modificación [39]. En el contexto de la computación en la nube, la portabilidad se puede clasificar en tres categorías: portabilidad de datos, portabilidad de funciones o aplicaciones y portabilidad de servicios o plataformas [40]. En particular, la portabilidad de la aplicación se refiere a la capacidad para definir las funcionalidades de la aplicación en un proveedor agnóstico manera.

Compatible con estándares abiertos como CAMP [41] y TOSCA [36] para modelar la topología de la aplicación y los ciclos de vida de los componentes facilitan el uso de CROF y aumenta aún más la reutilización de la topología definición, ya que restringe el problema de bloqueo del proveedor a nivel de proveedor de nube. La reutilización también se puede mejorar. a través de un enfoque modular con respecto a la aplicación descripción. Los métodos para lograr la modularidad incluyen plantillas, parametrización y herencia. Además, desde el esfuerzo inicial por describir aplicaciones y los componentes de la aplicación son altos, el intercambio de modelos por medio de bibliotecas o mercados existentes sería beneficioso.

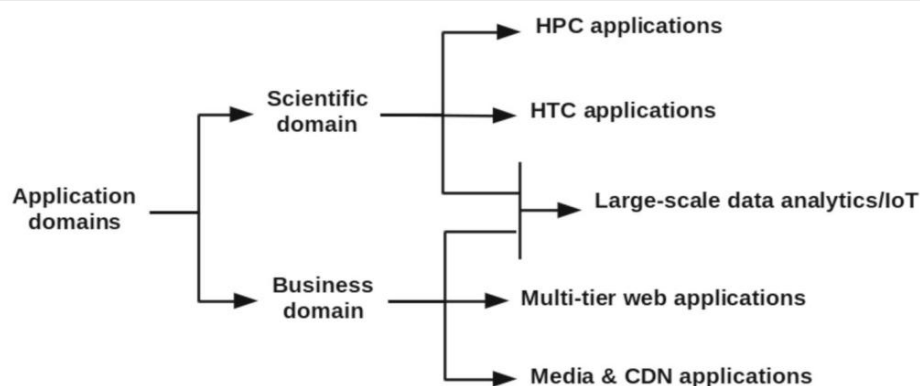
## Contenedorización

La virtualización basada en contenedores [42] es un enfoque clave para compartir el kernel del sistema operativo del host a través de múltiples instancias de invitados (es decir, contenedores), manteniéndolos aislado. Los contenedores de nivel de entorno proporcionan un recurso mecanismo de aislamiento con poca sobrecarga en comparación con Hipervisores a nivel de sistema operativo [43]. Además, el mayor aislamiento que ofrecen los contenedores permite reducir el consumo de recursos. configurarse, controlarse y limitarse a nivel de instancia.

<sup>13</sup><https://jclouds.apache.org/>

<sup>14</sup><http://libcloud.apache.org/>





**Fig. 5** Clasificación del dominio de aplicación

Docker es la plataforma líder basada en Linux para desarrollar, enviar y ejecutar aplicaciones a través de contenedores. virtualización basada.

Dado que manejar una gran cantidad de contenedores dentro de un El clúster de Docker puede ser una organización difícil, centrada en contenedores, como Docker Swarm<sup>15</sup>, Google Kubernetes<sup>16</sup>, y Apache Mesos<sup>17</sup> han aparecido. Ellos realizan orquestación a nivel de contenedor mediante la automatización del aprovisionamiento y la gestión de contenedores complejos implementaciones en varios hosts y ubicaciones.

#### Selección de recursos

La selección de recursos se refiere al nivel de automatización que admiten los CROF con respecto a la selección de hardware. y recursos de software. Suele implicar la identificación y análisis de recursos alternativos en la nube basados en la selección criterios. Los enfoques de selección de recursos se pueden clasificar en cuatro categorías.

En un enlace manual, los usuarios proporcionan el concreto único identificadores de las entidades de la nube. En un enlace automático especifican requisitos abstractos (por ejemplo, número de núcleos), cuales CROFs son responsables de unir a un concreto oferta en tiempo de ejecución. La vinculación automática se puede mejorar mediante ofreciendo un enlace optimizado, que aprovecha los criterios de optimización basados en los atributos del proveedor de la nube (p. ej., precio, ubicación) para seleccionar la oferta más adecuada. Una dinámica vinculante ofrece un sistema de resolución que permite cambios en el enlace basado en información de tiempo de ejecución (por ejemplo, datos métricos del sistema de monitoreo).

#### control del ciclo de vida

El control del ciclo de vida define las acciones que deben ejecutarse para administrar completamente las aplicaciones en la nube. Existente Los CROF proporcionan distintos niveles de automatización, normalmente categorizados como enfoques basados en scripts y DevOps.

Un enfoque basado en secuencias de comandos consiste en un conjunto de secuencias de comandos de shell, que se ejecutan en un orden específico. Tiene capacidad limitada para expresar dependencias, reaccionar a cambios y verificar configuraciones. Los enfoques basados en scripts se pueden extender a admitir herramientas DevOps (p. ej., Chef<sup>18</sup>, Puppet<sup>19</sup>, Ansible<sup>20</sup>) que ofrecen un enfoque más sofisticado para la implementación administración y descripciones de implementación listas para usar.

#### Cableado y flujo de trabajo

La mayoría de las aplicaciones en la nube se distribuyen con componentes residen en diferentes máquinas virtuales. Cuando la aplicación tiene lugar el despliegue, una instancia de aplicación que consiste de una o más instancias de componentes se crea. Ya que pueden existir relaciones de dependencia entre los componentes, la funcionalidad de implementación también tiene la tarea de cablear instancias de componentes juntos.

Un enfoque sencillo para resolver esas dependencias es el paso de atributos y eventos, en cuyo caso el ciclo de vida los scripts bloquean/esperan a que los atributos estén disponibles o registrar oyentes en eventos de cambio de topología. Una mejora es un flujo de trabajo manual definido por los usuarios para cuidar el orden de despliegue. Sin embargo, la forma más fácil para que los usuarios implementen aplicaciones es una automática. deducción del flujo de trabajo de las acciones del ciclo de vida definidas en componentes y sus relaciones. Además, los CROF puede ofrecer extensiones para servicios externos como IaaS/PaaS servicios (consulte la sección "Integración" ) para garantizar que el El motor de implementación es consciente de esta dependencia.

#### Supervisión

El seguimiento del comportamiento de las aplicaciones es la clave para evaluar la calidad de la implementación y un componente importante para la adaptación. Como primer paso esto comprende la recopilación de métricas. Los CROF deberían ofrecer una forma de medir las métricas del sistema (por ejemplo, el uso de la CPU) y la aplicación métricas (por ejemplo, número de solicitudes). Si métricas predefinidas

<sup>15</sup><https://docs.docker.com/engine/swarm/>

<sup>16</sup><https://kubernetes.io/> <sup>17</sup><http://mesos.apache.org/>

<sup>18</sup><https://www.chef.io/>

<sup>19</sup><https://puppet.com/>

<sup>20</sup><https://www.ansible.com/>

no son suficientes, una forma bien definida de agregar métricas personalizadas debería ser provisto. Los mecanismos de agregación permiten calcule métricas de nivel superior y combine también múltiples métricas. El acceso a datos históricos también es deseable en

para apoyar una evaluación de alto nivel del seguimiento datos.

#### Adaptación del tiempo de ejecución

Los CROF deben adaptar automáticamente las aplicaciones para para hacer frente a las desviaciones dinámicas (por ejemplo, aumento de la carga). Las operaciones para hacer frente a tales cambios son principalmente escalar, y migración. Sin embargo, el apoyo a la adaptación de muchos CROF se limita a la escala horizontal con disparadores basados en umbrales. Motores de reglas que aprovechan métricas complejas y los objetivos de QoS serían una mejora.

Dado que las implementaciones entre nubes pueden experimentar fallas, Los CROF también deben respaldar la recuperación de estados erróneos no deseados. Otra característica relacionada con la adaptación es la integración continua/entrega continua (CI/CD), que permite modificar el modelo de topología de las aplicaciones desplegadas reduciendo los cambios al mínimo posible.

#### Revisión de los CROF

Esta sección presenta una selección de CROF de diferentes paisajes. Sin perjuicio de que el estado actual del arte abarca un gran número de marcos, este obra contempla un subconjunto de ellos que consideramos ser representativo de las características de la mayoría de soluciones existentes. Clasificamos los marcos en dos categorías: CROF de producción/comerciales y experimentales/académicas.

Los CROF de producción/comerciales se utilizan en un entorno de producción por parte de proveedores de nube pública y privada. Mientras que algunos de ellos son de código cerrado, otros son de código abierto y respaldado por una comunidad próspera de desarrolladores y usuarios. CROF experimentales/académicos generalmente se originan en el escenario de investigación y avanzan el estado del arte, aunque su implementación es en su mayoría prototípicos.

Discutimos a continuación cada clase de CROF y analizamos sus principales capacidades desde la nube y la aplicación perspectivas, tal como se cubre ampliamente en “[Marco de análisis sección de trabajo](#)”. La tabla 1 proporciona una vista de pájaro de la marcos tomados en consideración. En concreto, cada fila representa un CROF (Nombre) y especifica los autores originales (Organización), las fechas básicas para el inicio y últimos lanzamientos (Activo), una breve introducción (Descripción), y las fuentes consultadas (Referencias).

#### CROF comerciales/de producción

Hoy en día, existe una gran variedad de CROFs de producción/comerciales alrededor [\[44\]](#), tales como servicios centrados en la infraestructura (p. ej., Heat, CloudFormation) proporcionados por proveedores de la nube que también son IaaS

proveedores, centrados en la plataforma (p. ej., Cloud Foundry, Open Shift) e independientes de la plataforma (p. ej., Cloudify, Terraform) herramientas de aprovisionamiento de recursos de proveedores de IaaS. En esto primeramente debatimos algunas de las soluciones más relevantes presentadas en la Tabla 1, y luego resumimos sus funciones de nube y aplicación en las tablas [2](#) y [3](#) respectivamente.

#### Calor

OpenStack Heat [\[45\]](#) es un servicio para gestionar la todo el ciclo de vida de la infraestructura y las aplicaciones dentro Nubes OpenStack. Implementa un motor de orquestación para lanzar múltiples aplicaciones en la nube compuestas basadas en un formato de plantilla compatible con CloudFormation (CFN) o en OpenStack Heat Orchestration nativo Formato de plantilla (HOT). Las plantillas HOT se definen en YAML.

Una plantilla de Heat describe la infraestructura de una nube aplicación de forma declarativa, lo que permite la creación de la mayoría de los tipos de recursos de OpenStack, así como más avanzados funciones (como alta disponibilidad de instancias, escalado automático de instancias y pilas anidadas) a través de OpenStack-native Llamadas a la API REST. Los recursos, una vez creados, son referidos como pilas. Las plantillas de calor son consumidas por Open StackClient, que proporciona una interfaz de línea de comandos (CLI) a las API de OpenStack para iniciar pilas, ver detalles de pilas en ejecución y actualizar y eliminar pilas

Heat solo permite una implementación de una sola nube en un Entorno OpenStack. Con referencia a la interoperabilidad, Heat no proporciona soluciones semánticas ni MDE, pero proporciona apoyo para TOSCA a través de la independiente Placas de proyecto Heat- [21](#) que traduce TOSCA tem Translator a HOT.

En cuanto a la portabilidad, Heat admite parcialmente el modelo estándares (TOSCA) y reutilización a través de parámetros de entrada, y composición de plantillas. También es compatible con la contenedorización mediante OpenStack Zun service [22](#).

Los recursos de la nube solo se pueden seleccionar de forma manual encuadración, mientras que los flujos de trabajo manuales y automáticos puede aprovechar las herramientas basadas en scripts o DevOps (como Chef y Puppet) para manejar toda la aplicación ciclo vital. Heat proporciona escalado horizontal con disparadores de umbral basados en métricas de infraestructura. parcialmente admite la entrega continua al actualizar las pilas existentes, dando como resultado que algunos recursos se actualicen en el lugar y otros siendo reemplazados con nuevos recursos. Las capacidades de recuperación de fallas también están respaldadas por medio de flujos de trabajo manuales y actualización de pilas.

[21https://wiki.openstack.org/wiki/Traductor de calor](https://wiki.openstack.org/wiki/Traductor_de_calor)

[22https://wiki.openstack.org/wiki/Zun](https://wiki.openstack.org/wiki/Zun)

[illegible]

Tabla 2 Comparación basada en la nube de CROF de producción/comerciales

Características de la nube	CROF						
	Calor	Cloudificar	brooklyn	estratos	alien4cloud	Terraformar	Formación de nubes
Multinube	x						x
nube cruzada	x						x
interoperabilidad							
- Estándares abiertos 00 x				x	0	x	x
- Semántica xx x				x	x	x	x
- MDE	xx x			x	x	x	x
- Bibliotecas y servicios abiertos	x	x			xxx		
Integración							
- Servicios IaaS/PaaS externos xx x				x	x		x
-POR EL	x			x		x	x
Acceso							
- Línea de comando							
- Tablero basado en la web							
- API basada en web							

x = no cumplido, 0 = parcialmente cumplido, = totalmente cumplido

Cloudificar

Cloudify [46] es un marco de orquestación de código abierto basado en TOSCA. Ofrece servicios para modelar aplicaciones y automatizar todo su ciclo de vida a través de un conjunto de flujos de trabajo integrados. Las plantillas de aplicación son denominados blueprints, que son documentos YAML escrito en DSL de Cloudify (lenguaje específico del dominio). Los blueprints normalmente son consumidos por la CLI de Cloudify, que incluye todos los comandos necesarios para ejecutar cualquier acciones en Cloudify Manager.

Los planos típicos contienen declaraciones para varios tipos de recursos, incluidos los recursos en la nube. Cloudify permite despliegues multinube y entre nubes por medio de complementos incorporados. También es compatible con BYON y aprovecha TOSCA para interoperabilidad y portabilidad. Sin embargo, a pesar de estar alineado con el estándar de modelado, el DSL de Cloud ify no hace referencia directa a los tipos estándar.

Cloudify admite la creación de contenedores mediante Docker. La orquestación de contenedores también está disponible a través de Kubernetes. Los recursos de la nube solo se pueden seleccionar de forma manual encuadernación, mientras que los flujos de trabajo manuales y automáticos puede aprovechar las herramientas basadas en scripts o DevOps (como Asi ble, Chef y Puppet) para manejar el ciclo de vida de la aplicación. Cloudify proporciona infraestructura, aplicaciones y métricas personalizadas. También permite la definición de agregaciones personalizadas y políticas usando Clojure23 y Riemann24.

Cloudify ofrece flujos de trabajo integrados para la aplicación curación (aplicando la lógica de los flujos de trabajo de desinstalación e instalación, respectivamente) y escalamiento horizontal. Escenarios complejos (p. ej., escalamiento vertical, estallido de nubes) no son compatibles fuera de la caja. Migración en vivo se cumple parcialmente en el contexto de contenedores aplicaciones, sin embargo. Se pueden mover varios pods con aplicaciones en contenedores entre nodos en el mismo clúster de Kubernetes, sin interrupción del servicio. La entrega continua se admite a través de actualizaciones de implementación, que permiten modificar una topología en ejecución agregando/eliminando/ modificando nodos. modificando los nodos existentes provocarán su reinstalación automática, aunque.

brooklyn

Apache Brooklyn [47] es un marco de código abierto para modelado, implementación y gestión de aplicaciones distribuidas definidas mediante planos YAML declarativos escritos en el DSL de Brooklyn. El formato YAML de Brooklyn sigue la especificación CAMP [41], pero utiliza algunos extensiones El apoyo a TOSCA está previsto para los próximos futuro. Los planos generalmente son consumidos por Brooklyn CLI del cliente para acceder a un servidor Brooklyn en ejecución. Una consola web y potentes REST-API están disponibles así como.

Brooklyn permite implementaciones multinube y entre nubes en muchas nubes públicas y privadas. también es compatible infraestructuras privadas (BYON), y aprovecha Apache jclouds como capa de abstracción de la nube para la interoperabilidad. La portabilidad se logra a través de mecanismos de reutilización modelo. (p. ej., herencia de tipos) y uso compartido de modelos (p. ej., tipos

23<https://clojure.org/>  
24<http://riemann.io/>

Tabla 3 Comparación basada en aplicaciones de CROF de producción/comerciales

CROF							
Características de la aplicación		Calor Cloudify Brooklyn Stratos Alien4Cloud Terraform CloudFormation					
Desarrollo	<b>Portabilidad</b>						
	- Modelo Normas 00 0			X	0	X	X
	- Modelo de reutilización	0	0	0			0
	- Compartir modelo	X		0			X
	<b>Containerización</b>		X				
	<b>Selección de recursos</b>						
	- Encuadernación manual						
	- Encuadernación automática xx 0			X	X	X	X
	- Encuadernación optimizada xx x			X	X	X	X
	- Encuadernación dinámica xx x			X	X	X	X
Despliegue	<b>Control del ciclo de vida</b>						
	- Basado en guiones			X			
	- DevOps						
	<b>Cableado y flujo de trabajo</b>						
	- Atributo y paso de eventos		0				
	- Flujo de trabajo manual		X			X	X
	- Flujo de trabajo automático		X	X			
	- Servicios Externos xx x			X	X		X
	<b>Supervisión</b>						
	- Métricas del sistema		X				
Ejecución	- Métricas de aplicación	X	X			X	
	- Métricas personalizadas	X		X		X	
	- Agregación		0				
	- Información histórica	X	X	X		X	
	<b>Adaptación del tiempo de ejecución</b>						
	- Basado en umbrales						
	- Motor de reglas	X	X			X	X
	- Escala horizontal						
	- Migración x0 x			0	0	X	0
	- Recuperación de fallas		0			X	
Ejecución	-CI/CD	0	0	X			

x = no cumplido, 0 = parcialmente cumplido, = totalmente cumplido

compartida localmente o en un repositorio de Git). brooklyn no admite contenedores fuera de la caja. Sin embargo, la contenedorización se puede integrar por medio de proyectos (por ejemplo, Cloudsoft Clocker25 ). Brooklyn admite manual y automático básico vinculante para la selección de recursos, mientras que no admite escenarios de flujo de trabajo. Acciones del ciclo de vida (es decir, efectores) para las entidades se pueden configurar a través de cualquiera de los scripts de shell

25<http://www.clocker.io/>

o Recetas del Chef. Brooklyn extrae métricas ya sea ejecutando acciones remotas o accediendo a un monitor externo herramienta. Sin embargo, es responsabilidad del usuario implementar esas acciones, o proporcionar una interfaz a un herramienta de monitoreo externo. Las métricas/QoS se pueden incorporar a las políticas, que toman acciones automáticamente, como el reinicio fallido nodos o escalamiento horizontal. De forma predeterminada, está disponible una política basada en umbrales. La entrega continua es exclusivamente posible a nivel de componente, a saber

mediante la redistribución de componentes individuales con actualización software.

### estratos

Apache Stratos [48] es un marco PaaS de código abierto que permite a los desarrolladores crear aplicaciones y servicios distribuidos. Las aplicaciones suelen estar compuestas de conjuntos de cartuchos que representan descripciones de resumen Máquinas virtuales que alojan servicios empresariales y de infraestructura, combinado con políticas de implementación y escalado. estratos define configuraciones y aplicaciones en un JSON específico formato, por lo que se pueden compartir. La reutilización es limitada, ya que los cartuchos contienen referencias a ID de IaaS Instantáneas y configuración de hardware. Las aplicaciones pueden gestionarse mediante Stratos CLI. Una consola web y También se encuentran disponibles potentes REST-API.

Stratos admite múltiples proveedores y utiliza Apache jclouds como capa de abstracción de la nube para la interoperabilidad.

A pesar de usar jclouds, BYON no es compatible. Tampoco se admiten servicios externos. Stratos aprovecha Kuber Netes como un marco de orquestación de clústeres para

proporcionar contenedorización. Los recursos de la nube son manualmente seleccionado al configurar los cartuchos. Además, mientras la descripción del ciclo de vida para administrar máquinas virtuales se realiza mediante Stratos, la configuración del software se delega a Puppet. Solo se admiten flujos de trabajo manuales.

Stratos usa un agente de cartucho que reside dentro de cada VM para acceder a las métricas del sistema y de la aplicación. Está no es posible definir métricas personalizadas. Uso en vuelo solicitudes, promedio de carga y métricas de memoria libre combinadas con un procesador de eventos complejo y Drools rule engine<sup>26</sup>, Stratos promulga una estrategia horizontal multifactorial escalado automático. También incluye el estallido de nubes, lo que permite migre sin problemas aplicaciones entre nubes. Recuperación Se admiten acciones en caso de que algunas tareas dentro de máquinas virtuales de una topología de aplicación falla, al destruir automáticamente y recrear la instancia del cartucho afectado. Continuo la entrega no es compatible, ya que los usuarios deben anular la implementación aplicaciones antes de cambiar sus definiciones.

### alien4cloud

Alien4Cloud (Aplicación Lifecycle ENabler para la nube) [49] es una plataforma de código abierto que hace que la aplicación administración en la nube fácil para las empresas. aprovecha otros proyectos de código abierto existentes que ayudan a orquestar aplicaciones en la nube y se centran en aspectos de tiempo de ejecución (por ejemplo, Cloudify). En Alien4Cloud, plantillas de aplicaciones (planos) se modelan en TOSCA para permitir interoperabilidad y portabilidad. Los planos también pueden ser compartido entre los usuarios de la plataforma a través de un catálogo de TOSCA mantenido. Sin embargo, Alien4Cloud admite una versión ligeramente modificada versión de TOSCA Simple Profile.

La implementación de la aplicación se realiza a través de un orquestador en una ubicación configurada y administrada por un orquestador Alien4Cloud admite una serie de tractores orquestales (Cloudify, Puccini<sup>27</sup> y Marathon<sup>28</sup>) a través de complementos.

Las ubicaciones describen un rango de destino de implementación lógica desde nubes privadas/públicas hasta un conjunto de máquinas físicas (BYON), o incluso contenedores Docker (Kubernetes y Mesos). Las implementaciones multinube y entre nubes son soportado.

Los recursos de la nube solo se pueden seleccionar a través del enlace manual (sustitución de nodo), mientras que tanto el manual como el los flujos de trabajo automáticos pueden aprovechar los scripts o DevOps herramientas (como Ansible, Chef y Puppet) para manejar el ciclo de vida de la aplicación. En cuanto al seguimiento y adaptación en tiempo de ejecución, ya que Cloudify se puede utilizar como La solución de orquestación de back-end de Alien4Cloud, la misma se aplican consideraciones. En particular, Alien4Cloud admite escalado horizontal y entrega continua.

### Terraform

Terraform [50] es una infraestructura de código abierto como código herramienta para construir, cambiar y versionar infraestructuras de una manera independiente de la plataforma. Utiliza su propio lenguaje de configuración de alto nivel conocido como Hashicorp Configuration Language (HCL), u opcionalmente JSON, para detallar el configuración de la infraestructura. A pesar de no cumplir con cualquier estándar de modelo, HCL admite la reutilización a través de módulos y composición de módulos. Los módulos reutilizables también pueden compartirse a través del Registro de Terraform, así como otras fuentes (p. ej., GitHub, Bitbucket). Las configuraciones son normalmente consumido por la CLI de Terraform, pero Terraform Enterprise también proporciona un panel basado en la web y API REST.

Terraform puede administrar múltiples proveedores de nube y incluso dependencias entre nubes por medio de complementos especiales llamados proveedores. Los proveedores están disponibles para Docker contenedores y orquestación de contenedores, así como externos servicios en la nube (por ejemplo, Amazon RDS<sup>29</sup>). Sin embargo, no se proporciona soporte para BYON. Los recursos de la nube son manualmente seleccionados durante la configuración, mientras que las acciones del ciclo de vida pueden configurarse a través de proveedores que ejecutan scripts o ejecutando la gestión de configuración (Chef, Puppet, Salt). Solo se admiten flujos de trabajo automáticos.

Terraform aprovecha los proveedores para proporcionar capacidades de escalado automático con disparadores de umbral en las métricas del sistema recopiladas por los servicios de monitoreo (por ejemplo, Azure Monitor<sup>30</sup>, Amazon CloudWatch<sup>31</sup>). La entrega continua es compatible mediante la aplicación de actualizaciones de configuración, que permiten para agregar/eliminar/modificar recursos. Cuando los argumentos de recursos no se pueden actualizar en el lugar, el recurso existente

<sup>27</sup><https://github.com/tliron/puccini>

<sup>28</sup><http://mesos.apache.org/>

<sup>29</sup><https://docs.aws.amazon.com/rds/index.html>

<sup>30</sup><https://docs.microsoft.com/en-us/azure/azure-monitor/>

<sup>31</sup><https://docs.aws.amazon.com/cloudwatch/index.html>

<sup>26</sup><https://www.drools.org/>



será reemplazado por uno nuevo en su lugar. Sin acciones de recuperación son compatibles desde el primer momento, ya que cualquier error debe ser abordado manualmente.

### Formación de nubes

AWS CloudFormation [51] es un modelo basado en herramienta de infraestructura como código para administrar implementaciones de infraestructura de AWS. Todos los recursos y dependencias son declarados en una plantilla JSON o YAML, que CloudFormation utiliza como modelo para crear recursos de AWS. A

La colección de recursos administrados se llama pila. Aunque Las plantillas de CloudFormation no se ajustan a ningún modelo estándares, la reutilización es parcialmente compatible a través de la entrada parámetros y pilas anidadas. Las plantillas suelen ser consumidas por la consola de CloudFormation, las API de REST o CLI.

CloudFormation solo puede modelar y administrar AWS recursos. No se proporciona soporte para múltiples nubes proveedores o BYON. La creación de contenedores se admite de forma nativa a través de los recursos de Elastic Container Service (ECS)<sup>32</sup>. La orquestación de contenedores también es compatible con Elastic Kubernetes Service (EKS) <sup>33</sup> recursos también. Los recursos de la nube se seleccionan a través del enlace manual, mientras que las acciones del ciclo de vida se pueden configurar a través de scripts de datos de usuario o herramientas DevOps (Chef, Puppet). Solo se admiten flujos de trabajo automáticos.

CloudFormation proporciona capacidades de escalado automático mediante AWS Auto Scaling<sup>34</sup>, que utiliza dinámica escalado y escalado predictivo para escalar automáticamente recursos basados en las métricas de Amazon CloudWatch. Las métricas personalizadas para Auto Scaling de aplicaciones también se pueden definir. La migración en vivo se cumple parcialmente en el contexto de aplicaciones en contenedores. Por ejemplo, es posible para migrar correctamente las aplicaciones existentes de un trabajador grupo de nodos a otro. Se admite la entrega continua por actualizaciones de pila. Según el recurso y las propiedades que se actualicen, una actualización podría interrumpir o incluso reemplazar un recurso existente. Las acciones de recuperación son compatibles con la reversión automática de la pila existente en falla.

### CROF experimentales/académicos

En esta sección, inicialmente revisamos un conjunto de CROF experimentales/académicos significativos descritos en Tabla 1, y luego resúmalos de acuerdo a su características de la nube y la aplicación en las tablas 4 y 5 respectivamente. Además, repasamos brevemente otras iniciativas de investigación centradas únicamente en aspectos específicos de CROF.

### Nublador

Cloudiator [52, 53, 62] es una nube cruzada de código abierto framework de orquestación, que se basa en Apache jclouds con el fin de admitir muchas plataformas de nube pública y privada. Se puede acceder al componente principal de orquestación, a saber, Colosseum, a través de un cliente Java o un usuario basado en la web. interfaz, o una REST-API.

La descripción de la aplicación consta de componentes individuales, que se ensamblan para formar una aplicación completa.

Cada componente proporciona operaciones de interfaz (por ejemplo, bash scripts) para gestionar el ciclo de vida de los componentes. Se describen las dependencias entre los componentes de la aplicación.

a través de entidades de comunicación que unen los puertos provistos y puertos requeridos. A pesar de no cumplir con ningún estándares de modelado, los componentes de la aplicación son reutilizables a través de diferentes aplicaciones.

El corredor de recursos es responsable de automáticamente seleccionando la oferta de nube correcta (anteriormente descubierta por el motor de descubrimiento), dependiendo de los requisitos/restricciones deseados en la configuración de la máquina virtual. El motor de implementación adquiere la máquina virtual y envía la solicitud de instalación del componente al equipo remoto. agente del ciclo de vida, a saber, Lance. Lance ejecuta componente instancias dentro de los contenedores de Docker de forma predeterminada. Además, solo se admiten flujos de trabajo automáticos.

Las capacidades de escalado automático se proporcionan mediante de AXE, un motor de seguimiento y adaptación integrado en Cloudiator, que implementa reglas de escalabilidad que consisten en condiciones basadas en umbrales vinculadas a datos sin procesar o métricas compuestas. Las funciones de migración se cumplen parcialmente al admitir el acceso a la funcionalidad de migración en vivo de OpenStack. Las acciones de recuperación están respaldadas por el motor de recuperación, que detecta estados anormales de las entidades del sistema marcándolas como fallidas y aplica soluciones basadas en categorías de fallas. el mismo mecanismo se utiliza para representar cambios en los modelos (entrega continua).

### roboconf

Roboconf [54, 63] es un marco de orquestación escalable de código abierto para plataformas de múltiples nubes. Muchas IaaS proveedores (por ejemplo, OpenStack, AWS, Azure, vSphere), como así como los contenedores Docker y las implementaciones locales para hosts locales, son compatibles mediante el uso de complementos especiales. Roboconf soporta parcialmente la interoperabilidad por medio de Extensiones OCCl y una implementación de destino genérico basado en apache jclouds. Además, se puede acceder por medio de una consola basada en shell, o un usuario basado en web interfaz o una API REST.

Roboconf proporciona un DSL inspirado en CSS, que permite describir aplicaciones y sus entornos de ejecución de forma jerárquica. Una aplicación distribuida se ve como un conjunto de componentes, construyendo un gráfico acíclico que describe tanto la contención como las relaciones en tiempo de ejecución entre

<sup>32</sup><https://docs.aws.amazon.com/ecs/index.html>

<sup>33</sup><https://docs.aws.amazon.com/eks/> <sup>34</sup><https://aws.amazon.com/autoscaling/>

**Tabla 4** Comparación basada en la nube de CROF experimentales/de investigación

Características de la nube	CROF					
	Nublador	roboconf	INDIGO-DataCloud	MiCADO	MODANubes	MarNubes
<b>Multinube</b>						
<b>nube cruzada</b>						
<b>interoperabilidad</b>						
- Estándares abiertos	x	0		0x 0		
- Semántica	x	x	x xx x			
- MDE	x xx x					x
- Bibliotecas y servicios abiertos	0x x					
<b>Integración</b>						
- Servicios IaaS/PaaS externos	x		x	x		
-POR EL	x			x	x	
<b>Acceso</b>						
- Línea de comando	x			x		x
- Tablero basado en la web					x	
- API basada en web						

x = no cumplido, 0 = parcialmente cumplido, = totalmente cumplido

componentes, y un grupo de instancias de estos componentes. Las definiciones de componentes se pueden reutilizar a través de resumen tipos (facetas), importaciones y herencia.

Roboconf consta de varios módulos. El Deployment Manager (DM) está a cargo de instanciar y administrar máquinas virtuales y agentes remotos. Los agentes usan complementos (como Bash o Puppet) para manejar el ciclo de vida de las instancias de software. El DM y el los agentes se comunican entre sí a través de un servidor de mensajería asíncrono. El SoftwareInstanceManager es responsable de generar automáticamente el software gestión del ciclo de vida y supervisión de instancias de software ellos mismos.

Las capacidades de escalado automático se proporcionan mediante gestión autónoma implementada por el DM y el agentes remotos. Los agentes envían notificaciones al DM cuando se vinculan ciertas condiciones basadas en umbrales al sistema se cumplen las métricas. El motor de decisión del DM responde a esas notificaciones usando el imperativo correspondiente normas. La supervisión de las métricas de la aplicación aún debe ser dirigido. Tanto las migraciones de aplicaciones como las reversiones globales/por componente (implementación continua) son parte de hoja de ruta de Roboconf, pero no son compatibles con la caja todavía.

## INDIGO-DataCloud

INDIGO-DataCloud (INtegrating Distributed data Infrast ructures for Global Exploitation) [55, 56, 64] es una plataforma informática y de datos de código abierto dirigida a científicos comunidades, y aprovisionado a través de la nube y basado en Grid infraestructuras, así como sobre clústeres HTC y HPC.

El marco INDIGO-DataCloud ha sido desarrollado

dentro del proyecto homónimo financiado por la UE Programa Marco Horizonte 2020 [65].

El proyecto INDIGO-DataCloud amplió los existentes Soluciones PaaS para proporcionar distribución automática de aplicaciones y/o servicios sobre un conjunto híbrido y heterogéneo de infraestructuras IaaS. Algunas de las claves Los componentes de INDIGO PaaS incluyen: orquestador, administrador de infraestructura (IM), CloudProviderRanker, monitoreo, administrador de SLA (SLAM), servicios administrados/aplicación (MSA) Servicios de implementación y gestión de datos. los Orchestrator coordina el proceso de implementación de servicios y aplicaciones tanto en IaaS local como pública plataformas Se puede acceder a través de una interfaz de línea de comandos (Orchent), o un portlet basado en GUI, o una API REST.

El orquestador delega la implementación al IM, a OpenStack Heat o a los frameworks Mesos, basados en Plantillas TOSCA y una lista de proveedores clasificados por el CloudProviderRanker. El componente de Monitoreo recopila datos de monitoreo de los servicios principales de PaaS y infraestructura/servicios del cliente mediante sondeos específicos. El SLAM establece un acuerdo entre el cliente y proveedor sobre los objetivos de capacidad y calidad. Los datos Management Services proporciona una capa de abstracción para acceder al almacenamiento de datos de forma unificada y federada manera.

INDIGO-DataCloud es compatible con múltiples nubes y implementaciones entre nubes, así como la interoperabilidad mediante el aprovechamiento de estándares abiertos (OCCI, CDMI). Eso también promueve la portabilidad mediante la adopción de una extensión de TOSCA para describir aplicaciones y servicios. Nube los recursos se seleccionan y optimizan automáticamente por el CloudProviderRanker, dependiendo de los SLA y

Tabla 5 Comparación basada en aplicaciones de CROF experimentales/de investigación

	CROF					
	Características de la aplicación	Cloudiator	Roboconf	INDIGO-DataCloud	MiCADO	MODAClouds SeaClouds
Desarrollo	<b>Portabilidad</b>					
	- Normas modelo	x	x	0 0 x 0		
	- Modelo de reutilización					0
	- Compartir modelo				0	
	<b>Contenedorización</b>					x x
	<b>Selección de recursos</b>					
	- Encuadernación manual	x		x		x x
	- Encuadernación automática		x		x	
	- Encuadernación optimizada		x		x	
	- Encuadernación dinámica	x	x	xxx		
Despliegue	<b>Control del ciclo de vida</b>					
	- Basado en guiones			x	x	x
	- DevOps	x			x	x
	<b>Cableado y flujo de trabajo</b>					
	- Atributo y paso de eventos					
	- Flujo de trabajo manual	x	x	x xx x		
	- Flujo de trabajo automático					
	- Servicios externos	x		x	x	
	<b>Supervisión</b>					
	- Métricas del sistema					
Ejecución	- Métricas de aplicación xx x					
	- Métricas personalizadas		x	x xx x		
	- Agregación		x			
	- Información histórica	x	x			
	<b>Adaptación del tiempo de ejecución</b>					
	- Basado en umbrales					
	- Motor de reglas					
	- Escala horizontal					
	- Migración	0x0x				0
	- Recuperación de fallas		x			0
	-CI/CD		x	x		x

x = no cumplido, 0 = parcialmente cumplido, = totalmente cumplido

datos de monitoreo. Una solución de gestión de la configuración basado en roles de Ansible se adopta para llevar a cabo tanto la despliegue de la aplicación y la creación de las imágenes de Docker preconfiguradas. Solo los flujos de trabajo automáticos son soportado.

Las acciones en tiempo de ejecución, como el escalado horizontal y el manejo de fallas, son compatibles automáticamente con MSA Despliegue (basado en Apache Mesos), que utiliza el Servicio de escalado automático (basado en EC3/CLUES35)

para garantizar la elasticidad del clúster, Marathon36 y Chronos37 frameworks para manejar servicios de larga ejecución (LRS) y trabajos de aplicaciones, respectivamente.

Marathon también puede migrar servicios si surgen problemas. A pesar de que se adoptaron diferentes prácticas de DevOps para ambos los servicios básicos y las aplicaciones de usuario (por ejemplo, las compilaciones de cada imagen de la aplicación se activan una vez que se el cambio está comprometido con su repositorio), los cambios calientes en las implementaciones de aplicaciones no son compatibles desde el primer momento.

35<https://www.grycap.upv.es/clues/esp/index.php>

36<https://mesosphere.github.io/marathon/>  
37<https://mesos.github.io/chronos/>

## MiCADO

MiCADO (nivel de aplicación en la nube basada en microservicios) Dynamic Orchestrator [57] es un marco de escalado automático y orquestación de múltiples nubes de código abierto para Contenedores Docker, orquestados por Kubernetes (o alternativamente por Docker Swarm). El marco completo de MiCADO ha sido investigado e implementado en el COLA (Orquestación en la nube a nivel de aplicación) proyecto financiado por la Comisión Europea [66].

Los servicios principales de MiCADO se implementan en MiCADO Master, que está configurado como el nodo maestro de Kubernetes. y proporciona el Docker Engine, Occopus [67] (a escala máquinas virtuales), Prometheus38 (para monitoreo), Policy Keeper (para realizar la decisión sobre el escalamiento), y Peticionario (para proporcionar punto final de envío) microservicios. Durante la operación, Los trabajadores de MiCADO son instanciados a pedido y se unen al clúster gestionado por MiCADO Master.

MiCADO es compatible con múltiples nubes y entre nubes implementaciones en varias infraestructuras de nube públicas y privadas. También proporciona interoperabilidad y portabilidad mediante una descripción de aplicación basada en TOSCA Plantilla (ADT), que consta de tres secciones: a) la definición de las aplicaciones individuales que componen un Despliegue de Kubernetes, b) la especificación de la VM y c) la implementación de políticas de escalamiento tanto para VM y niveles de escalado de Kubernetes. Los ADT pueden ser consumidos por mediante un panel de control basado en la web o una API REST.

Los recursos de la nube se seleccionan manualmente al configurar las máquinas virtuales. El ciclo de vida de la aplicación es manejado por MiCADO mismo, que aprovecha Occopus y Kubernetes para gestionar máquinas virtuales y contenedores, respectivamente. Solamente se admiten flujos de trabajo automáticos. MiCADO permite escalado automático según la máquina virtual y las métricas del contenedor recopiladas por dos exportadores integrados en cada MiCADO trabajador: Prometheus Node Exporter39 y CAdvisor40.

Las políticas de escalado se pueden definir específicamente para las aplicaciones. Por último, las capacidades de entrega continua se admiten a través de "actualizaciones continuas" en las implementaciones de Kubernetes.

## MODANubes

MODAClouds (Enfoque Dirigido por Modelos para el diseño y ejecución de aplicaciones en múltiples Nubes) [58, 59] es una plataforma de tiempo de diseño y tiempo de ejecución de código abierto para desarrollar y operar aplicaciones multinube con calidad de servicio garantizada. El marco MODAClouds ha sido desarrollado dentro del proyecto homónimo financiado por el Comisión Europea [68].

La caja de herramientas de MODAClouds consta de tres principales componentes: Creator4Clouds, Venues4Clouds y Energizer4Clouds. Creator4Clouds es una plataforma en tiempo de diseño que permite diseñar aplicaciones multinube, realizar

evaluación de rendimiento y costo, y planifique la estrategia de implementación eligiendo los proveedores de servicios que mejor adaptarse a todos los requisitos comerciales y de QoS. Venue4Clouds es un sistema de apoyo a la decisión (DSS) para elegir la más proveedores de nube adecuados dependiendo de diferentes aspectos tales como arquitectura de aplicaciones, riesgo comercial, calidad y costo. Energizer4Clouds es una plataforma de tiempo de ejecución para implementar, administrar, monitorear y asegurar las operaciones de múltiples nubes servicios. Específicamente, el subcomponente Tower4Clouds es responsable de recopilar, analizar y almacenar información de monitoreo, mientras que el subcomponente SpaceOps4Clouds promulga la autoadaptación de la aplicación para cumplir objetivos y/o limitaciones predefinidos siempre que ocurren cambios.

MODAClouds es compatible con múltiples nubes y entre nubes implementaciones en proveedores de IaaS y PaaS. Aprovecha un enfoque MDE para respaldar la interoperabilidad entre los proveedores de la nube. En particular, MODA CloudML es un conjunto de extensiones UML que permite a los desarrolladores para modelar aplicaciones multinube a través de tres niveles de abstracciones: Modelos independientes de computación habilitados para la nube (CCIM), Modelos independientes del proveedor de la nube (CPIM) y Modelos específicos del proveedor de la nube (CPSM). Estos modelos facilitan la portabilidad, ya que en su mayoría son reutilizable Los recursos de la nube se pueden seleccionar automáticamente y optimizado a través de Venues4Clouds y SpaceDev4Clouds, y administrado a través de scripts de shell o Puppet. Solamente se admiten flujos de trabajo automáticos.

Dentro del entorno de tiempo de ejecución de MODAClouds, el El motor Models@Runtime es responsable de promulgar acciones de adaptación, como escalado y explosión de aplicaciones, migración de datos y aplicaciones, y entrega continua tanto a nivel de infraestructura como de componentes. Falla la recuperación es parcialmente compatible con la migración de datos y escenarios de escalamiento/explosión.

## MarNubes

SeaClouds (Gestión multinube SEamless Adaptive de aplicaciones basadas en servicios) [60, 61] es un código abierto plataforma para implementar y administrar componentes múltiples aplicaciones sobre nubes heterogéneas. El MarNubes marco ha sido investigado e implementado dentro el proyecto homónimo financiado por la Comisión Europea [69].

La arquitectura de SeaClouds comprende seis componentes principales: Dashboard, Discoverer, Planner, Deployer, Monitor y SLA Service. El Dashboard permite modelar aplicaciones (topología y requisitos). El Discoverer identifica las capacidades disponibles que ofrece la nube proveedores El Planificador recibe el AAM (Modelo de aplicación abstracto) del Tablero y crea un conjunto de ADP (Abstract Deployment Plan) cumpliendo con la aplicación requisitos A partir del plan seleccionado, se generará un modelo de aplicación desplegable (DAM) que contenga el

38<https://prometheus.io/>

39[https://github.com/prometheus/node\\_exporter](https://github.com/prometheus/node_exporter)

40<https://github.com/google/cadvisor>

información que necesita el Deployer (basado en Apache Brooklyn [47]) para implementar, configurar y ejecutar la aplicación. El Monitor recopila la infraestructura y la aplicación. métricas de nivel de los proveedores de nube objetivo para para verificar que se cumplan los requisitos de QoS. Y si no, se pueden desencadenar acciones de reconfiguración. El Servicio SLA hace cumplir políticas orientadas al negocio y acciones comerciales aplicar en caso de infracción.

SeaClouds es compatible con múltiples nubes y entre nubes implementaciones en proveedores de IaaS y PaaS. También promueve la interoperabilidad y la portabilidad al adoptar una representación basada en TOSCA para AAM y ADP, así como una descripción basada en CAMP para DAM. Nube los recursos son seleccionados y optimizados automáticamente por el Planificador. Los cambios en la unión también pueden ocurrir en caso de acciones de reconfiguración. Solo los flujos de trabajo automáticos son soportado.

SeaClouds permite acciones de reparación, como escalar horizontal y verticalmente los recursos de la nube, o reiniciar y sustitución de componentes defectuosos. También es compatible con la replanificación. para manejar los casos que no pueden ser resueltos por reparando En este proceso puede ocurrir una migración de los módulos de la aplicación. No se admite la entrega continua fuera de la caja.

## Otras iniciativas

En esta sección, revisamos brevemente una serie de otros enfoques de investigación derivados de proyectos relacionados de la UE que abordan, en diversos grados, la organización de nubes múltiples, la interoperabilidad y la portabilidad. Específicamente, algunos trabajos apuntan a la interoperabilidad semántica (es decir, moSAIC, cloud4SOA), algunos exploran los beneficios de redes de nube federadas (BEACON, ATMOSPHERE), mientras que otros se centran en la portabilidad de aplicaciones a través de no estándar (es decir, Claudia, BEST, ASCETIC, HARNESS), parcialmente estándar (es decir, soCloud) y nube totalmente estándar (es decir, CELAR, CloudLightning) modelado.

mOSAIC [70, 71] es una API y una plataforma de código abierto para múltiples nubes diseñadas y desarrolladas dentro de la proyecto homónimo [72]. Despliegue de aplicaciones y la portabilidad a través de múltiples nubes se facilita por medio de una API común y una abstracción de alto nivel de la nube recursos. mOSAIC también permite a los desarrolladores de aplicaciones para especificar los requisitos de recursos en términos de una nube ontología, mientras que la plataforma, utilizando un mecanismo de intermediación, realiza un proceso de emparejamiento para encontrar los servicios en la nube más adecuados. Al hacerlo, los desarrolladores puede posponer su decisión sobre la adquisición de la nube servicios hasta el tiempo de ejecución. Sin embargo, a pesar de que un modelo de programación basado en componentes independiente de la plataforma es utilizado, las aplicaciones deben implementarse aprovechando una de las API compatibles con el idioma (Java, Pitón).

Cloud4SOA [73, 74] es una solución basada en intermediarios de múltiples nubes desarrollada bajo el proyecto homónimo [75], que aborda la interoperabilidad semántica y la portabilidad desafíos en la capa PaaS. Es compatible con multiplataforma emparejamiento, gestión, seguimiento y migración de aplicaciones mediante la interconexión semántica de ofertas de PaaS heterogéneas. Similar a MOSAIC, Cloud4SOA introduce una ontología de nube que establece un conjunto de abstracciones entre diferentes ofertas de PaaS mientras expone una API estandarizada multi-PaaS para la aplicación perfecta implementación y gestión en diferentes plataformas en la nube. A pesar de ser independiente de las API específicas que se ofrecen por las ofertas subyacentes de PaaS, adaptadores que actúan como un software intermedio entre la API de Cloud4SOA y PaaS nativa Las API aún son necesarias.

El objetivo principal del proyecto BEACON [76] es desarrollar técnicas para federar recursos de red en la nube y para permitir un despliegue eficiente y seguro de federados aplicaciones en la nube. En concreto, el enfoque propuesto es para construir una capa de virtualización homogénea sobre redes físicas subyacentes heterogéneas, computación y infraestructuras de almacenamiento. Aprovechando la combinación de Federación en la nube, redes definidas por software (SDN), y tecnologías de virtualización de funciones de red (NFV), el proyecto ha entregado un diseño innovador de un Sistema de gestión de federaciones que actúa como un proveedor de servicios externo que se ocupa de los servicios de red federados entre varias nubes OpenStack federadas [77].

ATMOSPHERE [78] tiene como objetivo diseñar e implementar un marco y plataforma que se basan en virtualización ligera, recursos híbridos y sistemas federados de Europa y Brasil infraestructuras para desarrollar, construir, desplegar, medir y Desarrollar aplicaciones confiables y habilitadas para la nube. Organización y despliegue de topologías de aplicaciones complejas se logra a través del estándar TOSCA. En el contexto del proyecto, los socios desarrollaron una arquitectura de red federada [79] mediante la creación de una superposición de múltiples inquilinos Redes a través de diferentes sitios. El marco desarrollado ofrece servicios tales como distribución e inter-sitio migración de máquinas virtuales, administración de recursos y red administración.

Claudia [80] es un sistema de gestión de servicios que implementa una capa de abstracción que permite la despliegue y escalado del servicio dependiendo tanto de la infraestructura como del estado del servicio. A la inversa de mOSAIC y Cloud4SOA, cada servicio en Claudia está definido por su archivo de descripción de servicio (SDF) correspondiente, cuya sintaxis es basado en el estándar OVF, proporcionando así proveedor y portabilidad de la plataforma. Sin embargo, las extensiones especiales de OVF debe definirse para soportar la escalabilidad automática, personalización en tiempo de implementación y conectividad externa especificación.

OPTIMIS [81] es un conjunto de herramientas que aborda y optimiza todo el ciclo de vida del servicio sobre la base de aspectos

como la confianza, el riesgo, la ecoeficiencia y el coste, teniendo en consideración una serie de escenarios de nube, a saber, nube federación, multinube, nube híbrida. Sin embargo, con respecto a la multinube, la interoperabilidad con sistemas que no son de OPTIMIS los proveedores solo se pueden lograr mediante el uso de API y adaptadores externamente a los componentes OPTIMIS. De acuerdo a Modelo de programación OPTIMIS, cada servicio está definido como una colección de elementos centrales que se empaquetan junto con cualquier componente de software externo en imágenes de VM. De manera similar a Claudia, estas imágenes de VM se configuran mediante de un manifiesto de servicio basado en el estándar OVF, pero se requiere un conjunto de extensiones OVF para especificar los requisitos funcionales y no funcionales del Servicio.

ASCETIC [82] es una arquitectura abierta y un enfoque para multi-nube optimizando la eficiencia energética, diseñado dentro el proyecto homónimo de la UE [83]. De manera análoga a OPTIMIS, la especificación OVF se emplea para definir un conjunto completo de máquinas virtuales que se implementarán en un proveedor de IaaS. Sin embargo, las extensiones OVF son necesarias para apoyar la negociación de SLA y las reglas de autoadaptación.

El proyecto HARNESS [84] desarrolla una plataforma de computación en la nube que incorpora recursos computacionales, de red y de almacenamiento heterogéneos y no tradicionales.

en la pila del centro de datos para proporcionar un alto rendimiento a bajo costo HARNESS prevé una PaaS en la nube mejorada pila de software que no solo es compatible con los productos básicos existentes pero también incorpora tecnologías heterogéneas como Dataflow Engines (DFE), programables enrutadores y diferentes tipos de dispositivos de almacenamiento [85]. los El proyecto demostró sus resultados a través de extensiones a Open Stack.

soCloud [86] es un servicio basado en componentes PaaS para gestionar portabilidad, elasticidad, aprovisionamiento, y alta disponibilidad en varias nubes. Solicitud Los descriptores se basan en el estándar OASIS Service Component Architecture (SCA) [87]. Sin embargo, desde el modelo SCA no permite definir no funcionales requisitos, se requieren extensiones SCA especiales. También se utiliza un DSL personalizado para describir la elasticidad. Además, soCloud no solo es compatible con sistemas basados en SCA aplicaciones, pero manteniendo las asignaciones a varios proveedores de la nube y mantenerse al día con las características recientes de las nubes soportadas son una preocupación.

CELAR [88, 89] es una plataforma de gestión de recursos capaz para implementar, monitorear y escalar aplicaciones automáticamente sobre una infraestructura en la nube. Las aplicaciones se describen utilizando TOSCA, que garantiza la portabilidad de las descripciones de las aplicaciones en diferentes plataformas IaaS. Sin embargo, cada vez que se va a implementar una nueva aplicación, los usuarios deben emitir la solicitud a la CELAR correspondiente Instancia de servidor dentro de la nube que quieren implementar su aplicación a. A diferencia de mOSAIC, cloud4SOA y ASCETIC, no se define ningún mecanismo de intermediación para

para adaptarse mejor a los requisitos de recursos de la nube. Es más, la nube cruzada no es compatible.

CloudLightning [90] es un servicio en la nube heterogéneo modelo de gestión y entrega desarrollado dentro del proyecto homónimo de la UE [91]. Basado en los principios de autoorganización y autogestión, CloudLightning permite a los usuarios diseñar y desplegar sus aplicaciones sin necesidad de seleccionar la más adecuada recursos. Esta separación de preocupaciones se hace posible mediante una descripción de servicio específica de CloudLightning (CL-SDL), que amplía TOSCA para capturar atributos específicos El enfoque declarativo es enriquecido con mecanismos de descubrimiento de recursos que permiten identificación y consumo más fácil de una variedad de recursos heterogéneos. CloudLightning propone una solución basada en un Servicio Gateway, que se basa en dos herramientas de código abierto: Alien4Cloud actuando como puerta de enlace Service UI y Brooklyn-TOSCA41 actuando como orquestador de implementación. En vista de lo anterior, las mismas observaciones realizados en la sección "Brooklyn" son aplicables a CloudLightning.

## Discusión crítica

Las tablas 2, 3, 4 y 5 resumen los CROF presentados en la sección "Revisión de los cROF" describiendo las características debatido en la sección "Marco de análisis". Discutimos el características principales de estos marcos a continuación.

La mayoría de los CROF revisados brindan acceso diferente modos, incluidos tableros basados en la web y API, y permitir despliegues tanto en múltiples nubes como entre nubes, excepto Heat y CloudFormation que, como servicios centrados en la infraestructura, solo admiten sus propios Proveedores de IaaS (es decir, OpenStack y Amazon, respectivamente). Además, algunos de ellos admiten implementaciones nativas en BYON (p. ej., Cloudify, Brooklyn, Roboconf, nubes marinas). La interoperabilidad entre los proveedores de la nube es logrado principalmente por medio de estándares abiertos y bibliotecas/capas de abstracción (por ejemplo, jclouds). Estándares abiertos parecen estar ganando terreno, especialmente en escenarios académicos. Como tal, varios CROF académicos brindan interoperabilidad a través de OCCl (p. ej., Roboconf) o CDMI. (por ejemplo, INDIGO-DataCloud), mientras que otros lo hacen a través de TOSCA (MiCADO, SeaClouds). A pesar de ser el foco de esfuerzos de investigación anteriores (p. ej., mOSAIC, Cloud4SOA), los enfoques semánticos parecen haber dejado de ser una prioridad en comparación con la adopción de estándares abiertos. De todo las iniciativas, MODAClouds es el único que emplea metodologías basadas en modelos.

Con respecto a la portabilidad de aplicaciones, los CROF de tanto la industria como la academia están colocando cada vez más importancia en los estándares de modelado. Sin embargo, mientras toma TOSCA (p. ej., Cloudify, MiCADO) y CAMP (p. ej.,

<sup>41</sup><https://github.com/cloudsoft/brooklyn-tosca>



Brooklyn, SeaClouds) como modelos de referencia, suceden para personalizar y ampliar los tipos estándar. Así, un ulterior sería adecuado un esfuerzo para garantizar una mayor cumplimiento de las especificaciones antes mencionadas. Aparte de la adopción de modelos estándar, reutilización de modelos se fomenta mediante módulos compartidos localmente o de forma remota. Además, dado que los contenedores proporcionan mejores encapsulación de aplicaciones y abstracción de recursos, la mayoría de los CROF soportan tanto contenedores como contenedores orquestación.

En cuanto al aprovisionamiento de recursos, existen diferentes aspectos del asunto que deben ser considerados, tales como: selección, configuración y despliegue de recursos. En escenarios de múltiples nubes, la selección está lejos de ser una trivialidad tarea debido a la diversidad de las características de los servicios en la nube y QoS. Si bien la selección manual es compatible con el mayoría de CROF, selecciones automáticas y optimizadas son apoyados casi exclusivamente por CROF académicos. Los la selección optimizada aprovecha la QoS y los requisitos técnicos, y se lleva a cabo en función de información estática sobre la calidad del servicio proporcionada por los proveedores de la nube o mediante negociación dinámica de SLAs. Algunos proyectos de nubes múltiples (p. ej., INDIGO-DataCloud, MODAClouds, SeaClouds) brindan soporte para la gestión de SLA, incluso aunque los SLA de nubes múltiples no están cubiertos. Actualmente hay soporte limitado disponible para la selección dinámica (es decir, nubes marinas).

El despliegue de recursos puede ser manual o automático. Si bien la mayoría de los CROF comerciales admiten tanto el manual y flujos de trabajo automáticos, CROF académicos exclusivamente admite los automáticos. Utilizando modelos estándar como TOSCA, en su caso, resulta útil tanto para definir un flujo de trabajo personalizado y para generar uno automáticamente. Sin embargo, dado que los estándares actuales carecen de soporte para modelar la semántica relacionada con la creación de instancias de relaciones entre instancias de componentes, el cableado real de las instancias de componentes dependen de las capacidades ofrecidas por la CROF promulgando el despliegue. En esa nota, extensiones estándar en apoyo de cableado sofisticado en nivel de instancia sería deseable. En cuanto a la configuración de recursos, por un lado, los scripts son ampliamente compatibles, pero por otro lado las herramientas de gestión de configuración en su mayoría son apoyados por CROF comerciales. No obstante, algunos proyectos académicos (p. ej., INDIGO-DataCloud y MODAClouds) explotan estas herramientas para promulgar Prácticas DevOps también.

El monitoreo juega un papel clave en el seguimiento del estado de aplicaciones y recursos físicos y virtuales. Monitoreo de métricas en diferentes niveles de abstracción (por ejemplo, de infraestructura y de aplicación) y la captura de dependencias entre estos niveles permite realizar análisis de causa raíz. análisis, de modo que cualquier problema a nivel de infraestructura pueda conducir automáticamente a la adaptación de la infraestructura en tiempo de ejecución que mejor se adapte a los requisitos de la aplicación en tiempo de ejecución. Tiempo

Las métricas de infraestructura son ampliamente compatibles con CROF comerciales y académicos, aplicaciones y personalizados.

Las métricas requieren una mayor investigación. Los mecanismos de agregación métrica están disponibles para una gran mayoría de CROF. Sin embargo, a la luz de los escenarios de nubes múltiples, donde las aplicaciones y los recursos pueden distribuirse en gran medida, es necesario recopilar y agregar métricas de entornos de nube heterogéneos. Como resultado, se deben inspeccionar las interfaces y los formatos estandarizados.

Los datos de seguimiento permiten diferentes propósitos tales como hacer cumplir los SLA, permitir la elasticidad, garantizar la QoS. Los SLA se pueden utilizar como base para servicios en la nube y respectivas aplicaciones a gestionar durante su ciclo de vida. La gestión de múltiples nubes requiere mecanismos específicos para la adaptación del tiempo de ejecución en una diversidad de configuraciones en la nube, incluida la escalabilidad, la migración, la tolerancia a fallas y la entrega continua. Mientras que los enfoques reactivos adaptación al tiempo de ejecución están bastante consolidados entre todos los CROF, los enfoques predictivos (basados en modelos de predicción de carga de trabajo y optimización de aprendizaje automático) solo se admiten en algunos CROF (por ejemplo, AWS CloudFormation) y debería ser más explorado

Tanto los CROF académicos como los comerciales proporcionan en gran medida compatibilidad con el escalado horizontal basado en umbrales. Los enfoques basados en políticas, especialmente en el panorama académico, también están ganando importancia. Soporte de migración todavía es limitada tanto en la industria como en la academia, ya que es estrechamente ligado a la portabilidad en todas sus facetas, es decir, portabilidad de VM, portabilidad de aplicaciones, portabilidad de datos. Aunque Los estándares independientes de la plataforma (TOSCA) y las técnicas de virtualización (contenedores) han mejorado la aplicación. encapsulación y abstracción de recursos, representación de datos independiente de la plataforma y estandarización de es necesario inspeccionar la importación y exportación de datos a través de nubes diversas y heterogéneas. A este respecto, MODAClouds proporciona una solución al problema de la migración de datos, aunque en el contexto de NoSQL escalable bases de datos

Tanto los CROF académicos como los comerciales respaldan el fracaso fallaron los mecanismos de recuperación basados en reiniciar/reemplazar componentes o, en el peor de los casos, retroceder pilas de aplicaciones completas. De todos los CROF académicos, Cloudi ator, MODAClouds y SeaClouds permiten identificar estados anormales e indeseables del sistema y aplicar un conjunto limitado de acciones autonómicas. Sin embargo, la aparición de configuraciones de nubes múltiples descentralizadas que conectan un una variedad más amplia de entidades y recursos requiere autonomía sistemas de gestión que consideran la autoorganización, la autogestión y la autorreparación en una diversidad de servicios en la nube implementaciones. La entrega continua está bien soportada en el panorama comercial, y también está ganando terreno en el académico debido al uso cada vez mayor de DevOps metodologías.

Conclusión

La tecnología de computación en la nube ha evolucionado mucho en los últimos años, transformando los recursos tradicionales de infraestructura, plataforma y software en componentes virtuales elásticos y bajo demanda. Sin embargo, los recursos heterogéneos y de múltiples capas deben organizarse de manera eficaz para garantizar que los usuarios finales reciban niveles de calidad aceptables.

En este trabajo, analizamos minuciosamente el panorama de orquestación de la nube: después de presentar una taxonomía de características y dimensiones relevantes, mapeamos y evaluamos varios marcos de trabajo de orquestación de recursos de la nube, enfocándonos especialmente en las capacidades de múltiples nubes. Este análisis sistemático ha permitido identificar temas clave de investigación abiertos, proponiendo también un conjunto de direcciones futuras de investigación en el escenario de la orquestación de la nube.

abreviaturas

CROF: marco de orquestación de recursos en la nube; MDE: impulsado por modelo Ingeniería; IaaS: Infraestructura como Servicio; PaaS: Plataforma como Servicio; CMP: plataforma de gestión en la nube; SLA: Acuerdo de Nivel de Servicio; TOSCA: especificación de topología y orquestación para aplicaciones en la nube; máquina virtual: máquina virtual; BYON: Traiga su propio nodo; CI/CD: Integración Continua/Continuo Entrega

Agradecimientos No aplicable.

Contribuciones de los

autores Todos los autores contribuyeron por igual al artículo. El(los) autor(es) leyeron y aprobaron el manuscrito final.

Financiamiento No aplicable.

Disponibilidad de datos y materiales No aplicable.

Conflicto de intereses

Los autores declaran que no tienen conflictos de intereses.

Recibido: 14 abril 2020 Aceptado: 4 agosto 2020  
Published online: 10 September 2020

Referencias

1. RightScale (2019) Informe sobre el estado de la nube de RightScale 2019. <https://info.flexera.com/SLO-CM-WP-State-of-the-Cloud-2019>. Consultado el 12 de octubre de 2019

2. Ranjan R, Benatallah B, Dustdar S, Papazoglou MP (2015) Programación de orquestación de recursos en la nube: descripción general, problemas y direcciones. *Cómputo de Internet IEEE* 19(5):46–56. <https://doi.org/10.1109/MIC.2015.20>

3. Weerasiri D, Barukh MC, Benatallah B, Sheng QZ, Ranjan R (2017) A Taxonomía y Encuesta de Técnicas de Orquestación de Recursos en la Nube. *ACM Comput Surv* 50(2):26–12641. <https://doi.org/10.1145/3054177>

4. Baur D, Seybold D, Griesinger F, Tsitsipas A, Hauser CB, Domaschka J (2015) Funciones de orquestación en la nube: ¿Son las herramientas adecuadas para su propósito? En: 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC). págs. 95–101. <https://doi.org/10.1109/UCC.2015.25>

5. Bousselmi K, Brahm Z, Gammoudi MM (2014) Servicios en la nube orquestación: un estudio comparativo de los enfoques existentes. En: 28th International Conference on Advanced Information Networking and Applications Workshops. págs. 410–416. <https://doi.org/10.1109/WAINA.2014.72>

6. Grozev N, Buyya R (2014) Arquitecturas y aplicaciones entre nubes corretaje: taxonomía y encuesta. *Softw Pract Experience* 44(3):369–390. <https://doi.org/10.1002/spe.2168>

7. Ferry N, Rossini A, Chauvel F, Morin B, Solberg A (2013) Hacia Aprovechamiento, implementación, monitoreo y adaptación basados en modelos de sistemas de nubes múltiples. En: IEEE Sexta Conferencia Internacional sobre Computación en la Nube. págs. 887–894. <https://doi.org/10.1109/NUBE.2013.133>

8. Petcu D (2014) Consumo de recursos y servicios de múltiples nubes. *Cómputo de cuadrícula J* 12(2):321–345. <https://doi.org/10.1007/s10723-013-9290-3>

9. Petcu D, Vasilakos A (2014) Portabilidad en las nubes: enfoques y oportunidades de investigación. *Experiencia de práctica informática escalable* 15(3):251–270. <https://doi.org/10.12694/scpe.v15i3.1019>

10. Ferry N, Rossini A (2018) CloudMF: Gestión basada en modelos de aplicaciones de nubes múltiples. *ACM Trans Internet Technol* 18(2):16–24. <https://doi.org/10.1145/3125621>

11. Ferrer AJ (2016) Investigación entre nubes: Visión para 2020. *Procedia Comput Sci* 97:140–143. <https://doi.org/10.1016/j.procs.2016.08.292>

12. Buyya R, Srirama SN, Casale G, Calheiros R, Simmhan Y, Varghese B, Gelenbe E, Javadi B, Vaquero LM, Netto MAS, Toosi AN, Rodriguez MA, Llorente IM, Vimercati SDCD, Samarati P, Milojicic D, Varela C, Bahsoon R, Assuncao MDD, Rana O, Zhou W, Jin H, Gentzsch W, Zomaya AY, Shen H (2018) Un manifiesto para la computación en la nube de la generación futura: direcciones de investigación para la próxima década. *ACM Comput Surv* 51(5):105–110538. <https://doi.org/10.1145/3241737>

13. Lewis GA (2013) Papel de los estándares en la interoperabilidad de la computación en la nube. En: 46ª Conferencia Internacional de Hawái sobre Ciencias de Sistemas. págs. 1652–1661. <https://doi.org/10.1109/HICSS.2013.470>

14. Badger L, Bohn R, Chandramouli R, Grance T, Karygiannis T, Patt-Corner R, Voas E (2010) Casos de uso de computación en la nube. <https://www.nist.gov/itl/use-cases>. Consultado el 12 de octubre de 2019.

15. Ahronovitz M, et al. (2010) Informe técnico sobre casos de uso de computación en la nube, versión 4.0. [http://www.cloud-council.org/Cloud\\_Computing\\_Use\\_Cases\\_Whitepaper-4\\_0.pdf](http://www.cloud-council.org/Cloud_Computing_Use_Cases_Whitepaper-4_0.pdf). Consultado el 12 de octubre de 2019

16. Grupo de trabajo de administración distribuida (2010) Casos de uso e interacciones para administrar las nubes. [https://www.dmtf.org/sites/default/files/standards/documents/DSP-ISO103\\_1.0.0.pdf](https://www.dmtf.org/sites/default/files/standards/documents/DSP-ISO103_1.0.0.pdf). Consultado el 12 de octubre de 2019.

17. Zhang Z, Wu C, Cheung DWL (2013) Una encuesta sobre interoperabilidad en la nube: taxonomías, estándares y práctica. *SIGMETRICS Realice Eval Rev* 40(4):13–22. <https://doi.org/10.1145/2479942.2479945>

18. Stravoskoufos K, Preventis A, Sotiriadis S, Petrakis EGM (2014) Una encuesta sobre enfoques para la interoperabilidad y portabilidad de los servicios de computación en la nube. En: Actas de la 4.ª Conferencia internacional sobre ciencia de servicios y computación en la nube (CLOSER2014). págs. 112–117. <https://doi.org/10.5220/0004856401120117>

19. García ÁL, del Castillo EF, Fernández PO (2016) Standards for enable heterogeneous IaaS cloud federations. *Interfaces de soporte de computación* 47:19–23. <https://doi.org/10.1016/j.csi.2016.02.002>

20. Gartner (2018) Panorama competitivo: corretaje de servicios en la nube. <https://www.gartner.com/en/documents/3889023/competitive-landscape-cloud-service-brokerage>. Consultado el 12 de octubre de 2019.

21. Liu F, Tong J, Mao J, Bohn RB, Messina JV, Badger ML, Leaf DM (2011) NIST Cloud Computing Reference Architecture. <https://www.nist.gov/publications/nist-cloud-computing-reference-architecture>. Consultado el 12 de octubre de 2019

22. Elhabbash A, Samreen F, Hadley J, Elkhatib Y (2019) Corretaje en la nube: una encuesta sistemática. *ACM Comput Surv* 51(6):119–111928. <https://doi.org/10.1145/3274657>

23. Bernstein D, Ludvigson E, Sankar K, Diamond S, Morrow M (2009) Blueprint for the Intercloud - Protocolos y formatos para la interoperabilidad de computación en la nube. En: Actas de la Cuarta Conferencia Internacional sobre Internet y Aplicaciones y Servicios Web de 2009. *ICIW '09*. págs. 328–336. <https://doi.org/10.1109/ICIW.2009.55>

24. Foro mundial de tecnología entre nubes (2010) Casos de uso y requisitos funcionales para la computación entre nubes: un libro blanco. [http://www.gicft.jp/doc/GICTF\\_Whitepaper\\_20100809.pdf](http://www.gicft.jp/doc/GICTF_Whitepaper_20100809.pdf). Consultado el 12 de octubre de 2019.

25. Petcu D (2013) Nubes múltiples: Expectativas y enfoques actuales. En: Actas del Taller internacional de 2013 sobre aplicaciones multinube y nubes federadas. *Multinube '13*. ACM, Nueva York. págs. 1–6. <https://doi.org/10.1145/2462326.2462328>

26. Toosi AN, Calheiros RN, Buyya R (2014) Entornos informáticos en la nube interconectados: desafíos, taxonomía y encuesta. *ACM Comput Surv* 47(1):7–1747. <https://doi.org/10.1145/2593512>

27. Nogueira E, Moreira A, Lucrédio D, Garcia V, Fortes R (2016) Problemas en el desarrollo de aplicaciones en la nube interoperables: definiciones, conceptos,

- enfoques, requisitos, características y modelos de evaluación. *J Softw Eng Res Dev* 4(1):7. <https://doi.org/10.1186/s40411-016-0033-6> 28. Kaur K, Sharma DS, Kahlon KKS (2017) Enfoques de interoperabilidad y portabilidad en nubes interconectadas: una revisión. *ACM Comput Surv* 50(4):49–14940. <https://doi.org/10.1145/3092698> 29. Bellendorf J, Mann ZÁ (2018) Topología de la nube y orquestación usando
- TOSCA: una revisión sistemática de la literatura. En: Kritikos K, Plebani P, de Paoli F (eds). *Computación en la Nube y Orientada a Servicios*. págs. 207–215. [https://doi.org/10.1007/978-3-319-99819-0\\_16](https://doi.org/10.1007/978-3-319-99819-0_16)
30. Domaschka J, Griesinger F, Baur D, Rossini A (2015) Más allá de la mera Estructura de la aplicación Pensamientos sobre el futuro de las herramientas de orquestación en la nube. *Procedia Comput Sci* 68:151–162. <https://doi.org/10.1016/j.procs.2015.09.231>
31. Khoshkbarforoushha A, Wang M, Ranjan R, Wang L, Alem L, Khan SU, Benatallah B (2016) Dimensiones para evaluar los marcos de orquestación de recursos en la nube. *Computadora* 49(2):24–33. <https://doi.org/10.1109/MC.2016.56>
32. Clústeres de proyectos europeos sobre la nube (2015) Inter-cloud Challenges, Expectativas y problemas Documento de posición del clúster: hoja de ruta de investigación inicial y clasificación de proyectos. <https://eucloudclusters.wordpress.com/future-cloud>. Consultado el 12 de octubre de 2019
33. Clusters of European Projects on Cloud (2016) Inter-cloud Challenges, Expectations and Issues Cluster Position Paper: Research Roadmap Update. <https://eucloudclusters.wordpress.com/future-cloud>. Consultado el 12 de octubre de 2019
34. Clústeres de proyectos europeos sobre la nube (2017) Future Cloud Cluster Vision for 2030. <https://eucloudclusters.wordpress.com/future-cloud>. Consultado el 12 de octubre de 2019
35. GigaSpaces Research CloudifyTeam (2016) Gestión de la nube en la empresa: una descripción general de la orquestación frente a PaaS frente a CMP. <https://cloudify.co/blog/cloud-management-roundup-orchestration-paas-cmp/>. Consultado el 12 de octubre de 2019 36. OASIS (2013) Especificación de topología y orquestación para la nube
- Aplicaciones Versión 1.0. <http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html>. Consultado el 6 de julio de 2020 37. Buyya R, Vecchiola C, Selvi ST (2013) Capítulo 10 - Aplicaciones en la nube. En: Buyya R, Vecchiola C, Selvi ST (eds). *Dominar la computación en la nube*. Morgan Kaufmann, Boston. págs. 353–371. <https://doi.org/10.1016/B978-0-12-411454-8.00010-3>
38. Costache S, Dib D, Parlantzias N, Morin C (2017) Gestión de recursos en sistemas de plataforma en la nube como servicio: Análisis y oportunidades. *J Syst Softw* 132:98–118. <https://doi.org/10.1016/j.jss.2017.05.035> 39. Kolb S, Wirtz G (2014) Towards Application Portability in Platform as a
- Servicio. En: IEEE 8th International Symposium on Service Oriented System Engineering. págs. 218–229. <https://doi.org/10.1109/SOSE.2014.26>
40. Oberle K, Fisher M (2010) ETSI CLOUD - Estandarización inicial
- Requisitos para los Servicios en la Nube. En: *Actas de la 7ª Conferencia Internacional sobre Economía de Redes, Nubes, Sistemas y Servicios*. GECON'10. Springer, Berlín, Heidelberg. págs. 105–115. [https://doi.org/10.1007/978-3-642-15681-6\\_8](https://doi.org/10.1007/978-3-642-15681-6_8) 41. OASIS (2014) Gestión de aplicaciones en la nube para plataformas Versión 1.1.
- <http://docs.oasis-open.org/camp/camp-spec/v1.1/camp-spec-v1.1.html>. Consultado el 12 de octubre de 2019. 42. Soltesz S, Pötzl H, Fiuczynski ME, Bavier A, Peterson L (2007) Virtualización de sistemas operativos basada en contenedores: una alternativa escalable y de alto rendimiento a los hipervisores. En: *Actas de la 2ª Conferencia Europea ACM SIGOPS/EuroSys sobre Sistemas Informáticos 2007*. EuroSys'07. ACM, Nueva York. págs. 275–287. <https://doi.org/10.1145/1272996.1273025>
43. Singh S, Singh N (2016) Contenedores y Docker: Roles emergentes y futuro de la tecnología en la nube. En: *II Congreso Internacional de Tecnologías de la Comunicación y la Computación Teórica y Aplicada (iCATccT)*. págs. 804–807. <https://doi.org/10.1109/ICATCCT.2016.7912109> 44. Komarek A, Pavlik J, Sobeslav V (2017) Orquestación de sistemas híbridos con TOSCA y Salt. *J Eng Appl Sci* 12(9):2396–2401. <https://doi.org/10.36478/jeasci.2017.2396.2401> 45. OpenStack (2016) OpenStack Calor. <https://wiki.openstack.org/wiki/Heat>.
- Consultado el 12 de octubre de 2019
46. Cloudify (2019) Cloudify. <http://cloudify.co/>. Consultado el 12 de octubre de 2019
47. The Apache Software Foundation (2016) The Apache Brooklyn project. <https://brooklyn.apache.org/>. Consultado el 12 de octubre de 2019
48. Apache (2015) Apache Stratos. <https://stratos.apache.org/>. Consultado 12 octubre de 2019
49. FastConnect (2018) Alien4Cloud. <https://alien4cloud.github.io>. Consultado 12 de octubre
- de 2019 50. HashiCorp (2019) HashiCorp Terraform. <https://www.terraform.io/>. Consultado el 12 de octubre de 2019
51. Amazon (2016) Amazon CloudFormation. <https://aws.amazon.com/cloud/discretionary-for/discretionary-maldiscretionary-tion/>. Consultado el 12 de octubre de 2019
52. Baur D, Domaschka J (2016) Experiencias de la construcción de una nube cruzada herramienta de orquestación. En: *Actas del 3er Taller sobre Infraestructuras y Plataformas CrossCloud*. CrossCloud'16. págs. 4–146. <https://doi.org/10.1145/2904111.2904116> 53. Baur D, Seybold D, Griesinger F, Masata H, Domaschka J (2018) A Provider-Agnostic Approach to Multi-cloud Orchestration Using a Constraint Language. En: *18º Simposio Internacional IEEE/ACM sobre Clúster, Nube y Grid Computing (CCGRID)*. págs. 173–182. <https://doi.org/10.1109/CCGRID.2018.00032> 54. Pham LM, Tchana A, Donsez D, de Palma N, Zurczak V, Gibello P (2015)
- Roboconf: un orquestador de nube híbrida para implementar aplicaciones complejas. En: *IEEE 8th International Conference on Cloud Computing*. págs. 365–372. <https://doi.org/10.1109/NUBE.2015.56> 55. Salomoni D, Campos I, Gaido L, et al. (2016) Indigo-nube de datos:
- fundamentos y descripción arquitectónica de una plataforma como servicio orientada a la computación científica. *CoRRabs/1603.09536*. 1603.09536. Consultado el 6 de julio
- de 2020 56. Salomoni D, Campos I, Gaido L, et al. (2018) INDIGO-DataCloud: una Plataforma para facilitar el acceso sin problemas a las infraestructuras electrónicas. *Cómputo de cuadrícula J* 16(3):381–408. <https://doi.org/10.1007/s10723-018-9453-3> 57. Kiss T, Kacsuk P, Kovacs J, Rakoczi B, Hajnal A, Farkas A, Gesmier G, Terstyanszky G (2019) MiCADO: orquestador dinámico de nivel de aplicación en la nube basado en microservicios. *Futur Gener Comput Syst* 94:937–946. <https://doi.org/10.1016/j.future.2017.09.050> 58. Ardagna D, Di Nitto E, Mohagheghi P, Mosser S, Ballagny C, D'Andria F, Casale G, Matthews P, Nechifor C, Petcu D, Gericke A, Sheridan C (2012)
- MODAClouds: un enfoque basado en modelos para el diseño y la ejecución de aplicaciones en múltiples nubes. En: *4º Taller Internacional de Modelado en Ingeniería de Software (MISE)*. págs. 50–56. <https://doi.org/10.1109/MISE.2012.6226014>
59. Nitto ED, Matthews P, Petcu D, Solberg A (2017) Model-Driven Development and Operation of Multi-Cloud Applications: The MODAClouds Approach. [https://doi.org/10.1007/978-3-319-46031-4\\_60](https://doi.org/10.1007/978-3-319-46031-4_60) Brogi A, Carrasco J, Cubo J, D'Andria F, Ibrahim A, Pimentel E, Soldani J
- (2014) SeaClouds: Gestión multinube adaptable y perfecta de aplicaciones basadas en servicios. En: *17ª Conferencia Iberoamericana en Ingeniería de Software (CIBSE 2014)*. Curran Associates, Inc., Pucón. págs. 95–108 61. Brogi A, Fazzolari M, Ibrahim A, Soldani J, Wang P, Carrasco J, Cubo J,
- Durán F, Pimentel E, Di Nitto E, D'Andria F (2015) Gestión adaptativa de aplicaciones en múltiples nubes: el enfoque SeaClouds. *CLEI Electron J* 18:2–2. <https://doi.org/10.19153/cleiej.18.1.1> 62. Universidad de Ulm (2015) Cloudiator. <http://cloudiator.org/>. Consultado 12 octubre de 2019
63. Linagora (2013) Roboconf. <http://roboconf.net>. Consultado el 12 de octubre de 2019 64. Caballer M, Zala S, García ÁL, Moltó G, Fernández PO, Velten M (2018)
- Orquestación de arquitecturas de aplicaciones complejas en nubes heterogéneas. *Cálculo de cuadrícula J* 16(1):3–18. <https://doi.org/10.1007/s10723-017-9418-y>
65. Consorcio INDIGO (2017) El proyecto INDIGO-DataCloud. <https://www.indigo-datacloud.eu/>. Consultado el 12 de octubre de 2019
66. Consorcio COLA (2017) The COLA Project. <https://proyecto-cola.eu/>. Consultado el 12 de octubre de 2019
67. Kovács J, Kacsuk P (2018) Occopus: un orquestador de nubes múltiples para implementar y administrar infraestructuras científicas complejas. *Cómputo de cuadrícula J* 16(1):19–37. <https://doi.org/10.1007/s10723-017-9421-3> 68. Consorcio MODAClouds (2012) El proyecto MODAClouds. <http://multiclouddevops.com/>. Consultado el 12 de octubre de 2019.
69. Consorcio SeaClouds (2013) The SeaClouds project. <http://www.seaclouds-project.eu/>. Consultado el 12 de octubre de 2019 70. Petcu D, Macariu G, Panica S, Craciun C (2013) Portable Cloud
- aplicaciones: de la teoría a la práctica. *Futur Gener Comput Syst* 29 (6): 1417–1430. <https://doi.org/10.1016/j.future.2012.01.009>

71. Petcu D, Martino BD, Venticinque S, Rak M, Máhr T, Lopez GE, Brito F, Cossu R, Stopar M, Šperka S, Stankovski V (2013) Experiencias en la construcción de un mosaico de nubes. *J Cloud Comput Adv Syst Appl* 2(1):12. <https://doi.org/10.1186/2192-113X-2-12> 72. Consorcio mOSAIC (2010) El proyecto mOSAIC. <http://www.mosaiccloud.eu/>. Consultado el 12 de octubre de 2019
73. D'Andria F, Bocconi S, Cruz JG, Ahtes J, Zeginis D (2012) Cloud4SOA: Gestión de aplicaciones de múltiples nubes en las ofertas de PaaS. En: 14º Simposio Internacional sobre Algoritmos Simbólicos y Numéricos para Computación Científica. págs. 407–414. <https://doi.org/10.1109/SYNASC.2012.sesenta y cinco>
74. Kamateri E, Loutas N, Zeginis D, Ahtes J, D'Andria F, Bocconi S, Gouvas P, Ledakis G, Ravagli F, Lobunets O, Tarabanis KA (2013) Cloud4SOA: A Semantic-Interoperability PaaS Solution for Multi-Gestión y portabilidad de plataformas en la nube. En: Lau KK, Lamersdorf W, Pimentel E (eds). *Computación en la nube y orientada a servicios*, ESOC 2013. pp 64–78. [https://doi.org/10.1007/978-3-642-40651-5\\_6](https://doi.org/10.1007/978-3-642-40651-5_6)
75. Consorcio Cloud4SOA (2010) El proyecto Cloud4SOA. <http://www.cloud4soa.eu/>. Consultado el 12 de octubre de 2019
76. Consorcio BEACON (2015) El proyecto BEACON: Habilitación de redes federadas en la nube. <http://www.beacon-project.eu/>. Consultado el 12 de octubre de 2019
77. Celesti A, Levin A, Massonet P, Schour L, Villari M (2016) Federación servicios de red en múltiples nubes openstack. págs. 338–352. [https://doi.org/10.1007/978-3-319-33313-7\\_26](https://doi.org/10.1007/978-3-319-33313-7_26)
78. Consorcio ATMOSPHERE (2017) El proyecto ATMOSPHERE: Adaptive, Ecosistema híbrido fiable, gestionable, orquestado, seguro y que garantiza la privacidad para una informática en la nube resistente. <https://www.atmosphereeubrazil.eu/>. Consultado el 12 de octubre de 2019. 79. Castañeda IA, Blanquer I, de Alfonso C (2019) Facilitando el despliegue y la gestión de redes federadas en la nube en clústeres virtualizados. En: Actas de la 9.ª Conferencia internacional sobre ciencia de servicios y computación en la nube, CLOSER 2019, Heraklion, Creta, Grecia, 2 al 4 de mayo de 2019. págs. 601–608. <https://doi.org/10.5220/0007877406010608> 80. Roderio-Merino L, Vaquero LM, Gil V, Galán F, Fontán J, Montero RS, Llorente IM (2010) De la entrega de infraestructuras a la gestión de servicios en las nubes. *Futur Gener Comput Syst* 26 (8): 1226–1240. <https://doi.org/10.1016/j.future.2010.02.013> 81. Ferrer AJ, Hernández F, Tordsson J, Elmroth E, Ali-Eldin A, Zsigri C, Sirvent R, Guitart J, Badia RM, Djemame K, Ziegler W, Dimitrakos T, Nair SK, Kousiouris G, Konstanteli K, Varvarigou T, Hudzia B, Kipp A, Wesner S, Corrales M, Forgó N, Sharif T, Sheridan C (2012) OPTIMIS: un enfoque holístico para el aprovisionamiento de servicios en la nube. *Futur Gener Comput Syst* 28(1):66–77. <https://doi.org/10.1016/j.future.2011.05.022> 82. Ferrer AJ, Lordan F, Ortiz D, Guitart J, Macias M, Panuccio P, M. Badia R, Ponsard C, Temporale C, García D, Sirvent R, Deprez J, Sommacampagna D, Djemame K, Armstrong D, Agiatzidou E, Ejarque J, Blasi L, Kammer M (2014) Ascetic: adaptando el ciclo de vida del servicio hacia nubes eficientes. En: Espacio Europeo de Proyectos sobre Sistemas de Información y Comunicación - EPS Barcelona. SciTePress, Barcelona. págs. 89–106. <https://doi.org/10.5220/0006183400890106> 83. Consorcio ASCETIC (2013) El proyecto ASCETIC. <http://ascetic-project.Ue/>. Consultado el 12 de octubre de 2019 84. Consorcio HARNESS (2012) The HARNESS project: Hardware and
- Sistemas de software mejorados en red para computación en la nube. <http://www.arnés-proyecto.eu/>. Consultado el 12 de octubre de 2019. 85. Coutinho JGF, Pell O, O'Neill E, Sanders P, McGlone J, Grigoras P, Luk W, Ragusa C (2014) Proyecto Harness: Gestión de recursos informáticos heterogéneos para una plataforma en la nube. En: Goehring D, Santambrogio MD, Cardoso JMP, Bertels K (eds). *Computación reconfigurable: arquitecturas, herramientas y aplicaciones*. Saltador. págs. 324–329. [https://doi.org/10.1007/978-3-319-05960-0\\_36](https://doi.org/10.1007/978-3-319-05960-0_36) 86. Paraiso F, Merle P, Seinturier L (2016) soCloud: una solución orientada a servicios
- PaaS basada en componentes para gestionar la portabilidad, el aprovisionamiento, la elasticidad y la alta disponibilidad en varias nubes. *Informática* 98(5):539–565. <https://doi.org/10.1007/s00607-014-0421-x> 87. OASIS (2011) Arquitectura de componentes de servicio (SCA). <http://www.oasis-opencsa.org/sca>. Consultado el 12 de octubre de 2019
88. Giannakopoulos I, Papailiou N, Mantas C, Konstantinou I, Tsoumakos D, Koziris N (2014) CELAR: Plataforma de elasticidad de aplicaciones automatizadas. En: IEEE Conferencia Internacional sobre Big Data (Big Data). págs. 23–25. <https://doi.org/10.1109/BigData.2014.7004481> 89. Consorcio CELAR (2012) El proyecto CELAR. <http://www.celarccloudproject.eu/>. Consultado el 12 de octubre de 2019. 90. Selea T, Dragan I, Forti, s TF (2017) The CloudLightning Approach to Cloud-user Interaction. En: Actas del 1er Taller Internacional sobre la Próxima Generación de Arquitecturas en la Nube (NG-Cloud). *CloudLightning* (2018) El proyecto CloudLightning. <https://cloudlightning.eu/>. Consultado el 12 de octubre de 2019

## Nota del editor Springer

Nature se mantiene neutral con respecto a los reclamos jurisdiccionales en mapas publicados y afiliaciones institucionales.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)