



Ingeniería, investigación y tecnología  
ISSN: 1405-7743  
Facultad de Ingeniería, UNAM

Ruiz-Paz, Sergio Fabián; Santaolaya-Salgado, René; Fragoso-Díaz, Olivia G.; Álvarez-Rodríguez, Francisco Javier; Rojas-Pérez, Juan Carlos  
Modelo de orquestación dinámica para flujos de trabajo del software como servicio  
Ingeniería, investigación y tecnología, vol. XX, núm. 3, e034, 2019, Julio-Septiembre  
Facultad de Ingeniería, UNAM

DOI: <https://doi.org/10.22201/fi.25940732e.2019.20n3.034>

Disponible en: <https://www.redalyc.org/articulo.oa?id=40465052010>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org



## Modelo de orquestación dinámica para flujos de trabajo del software como servicio Dynamic orchestration model for software as a service workflows

Ruíz-Paz Sergio Fabián

Tecnológico Nacional de México, CENIDET

Centro Nacional de Investigación y Desarrollo Tecnológico

Departamento de Ciencias Computacionales, Cuernavaca

Correo: [friuz@cenidet.edu.mx](mailto:friuz@cenidet.edu.mx)

<https://orcid.org/0000-0001-6960-5460>

Santaolaya-Salgado René

Tecnológico Nacional de México, CENIDET

Centro Nacional de Investigación y Desarrollo Tecnológico

Departamento de Ciencias Computacionales, Cuernavaca

Correo: [rene@cenidet.edu.mx](mailto:rene@cenidet.edu.mx)

<https://orcid.org/0000-0003-3408-5818>

Fragoso-Díaz Olivia G.

Tecnológico Nacional de México, CENIDET

Centro Nacional de Investigación y Desarrollo Tecnológico

Departamento de Ciencias Computacionales, Cuernavaca

Correo: [ofragoso@cenidet.edu.mx](mailto:ofragoso@cenidet.edu.mx)

<https://orcid.org/0000-0001-8512-1941>

Álvarez-Rodríguez Francisco Javier

Universidad Autónoma de Aguascalientes

Departamento de Ciencias de la Computación

Correo: [fjalvar@correo.uaa.mx](mailto:fjalvar@correo.uaa.mx)

<https://orcid.org/0000-0001-6608-046X>

Rojas-Pérez Juan Carlos

Tecnológico Nacional de México, CENIDET

Centro Nacional de Investigación y Desarrollo Tecnológico

Departamento de Ciencias Computacionales, Cuernavaca

Correo: [carlos.rojas@cenidet.edu.mx](mailto:carlos.rojas@cenidet.edu.mx)

<https://orcid.org/0000-0002-0189-8532>

### Resumen

Uno de los modelos de despliegue con mayor uso de la Nube Computacional es el software como servicio, en el cual, el software se despliega como servicio a través de Internet para poder ser utilizado por los clientes en un modelo de pago por uso. Para poder atender de forma eficiente y dinámica a los clientes de la nube, este tipo de software se compone de un conjunto de servicios atómicos, o microservicios, alojados geográficamente en lugares diferentes, que han sido orquestados y coreografiados en un flujo de trabajo denominado Software como Servicio (SaaS) para representar un proceso de negocios. Bajo este tipo de composición el SaaS, en tiempo de ejecución, puede dar pie a fallos o comportamientos anormales en los resultados, lo que dificulta al proveedor de servicios identificar la falla para aplicar la reparación necesaria al problema y desplegar un SaaS funcional al cliente que lo requiere. Este artículo presenta un modelo de orquestación dinámica aplicado al flujo de trabajo del SaaS. El objetivo es anticipar las fallas que se puedan presentar en los servicios atómicos que lo conforman, en tiempo de ejecución, y orquestar dinámicamente los servicios atómicos con el grado de disponibilidad y confiabilidad necesaria para la ejecución exitosa del flujo de trabajo, asegurando con ello la ejecución en tiempo y forma del SaaS utilizado por el cliente. Este modelo fue implementado en el lenguaje Java con el objetivo de probar su funcionalidad.

**Descriptores:** Computación en la nube, software como servicio, orquestación dinámica, flujo de trabajo, servicios atómicos, micro-servicios.

### Abstract

One of the most used deployment model of the Cloud Computing is Software as a Service (SaaS), in which the software is deployed as a service through the Internet to be used by customers in a pay-per-use model. In order to serve efficiently and dynamically to customers of the cloud, this type of software is composed of a set of atomic services, or microservices, hosted geographically in different places, which have been orchestrated and/or and choreographed in a workflow called Software as a Service (SaaS) to represent a business process. Under this type of composition, the SaaS, in execution time, can give rise to failures or abnormal behaviors in the results. This make it difficult for the service provider to identify the failure to apply the necessary maintenance to repair the problem and deploying a functional SaaS to the client that requires it. This paper presents a model of dynamic orchestration applied to the workflow of SaaS. The aim is anticipating the faults that may occur in the atomic services that comprise it, in execution time, and dynamically orchestrate the atomic services with the degree of availability and reliability necessary for the successful execution of the workflow, thus ensuring the execution in time and form of the SaaS used by the client. The model was implemented in the Java language as a method with the objective of testing its functionality.

**Keywords:** Cloud computing, software as a service, dynamic orchestration, workflow, atomic service, microservices.

## INTRODUCCIÓN

La Nube Computacional se ha convertido en un nuevo paradigma tecnológico que está siendo ampliamente utilizado por la industria de las tecnologías de información y telecomunicaciones, debido a la facilidad que aporta en la oferta de servicios sobre demanda, su elasticidad, flexibilidad y suministro de recursos de cómputo. La Nube despliega una gama de soluciones tecnológicas al mercado, permitiendo ofertar y utilizar servicios informáticos ya sean públicos o privados, gratuitos o con algún costo de operación, con la finalidad de cumplir sus objetivos, mejorar su productividad, reducir costos, mejorar sus servicios, entre otros beneficios, facilitando con ello el acceso bajo demanda. Todo esto, sin la preocupación de que los servicios tengan que ser instalados en el ambiente local del usuario, eliminando el costo y esfuerzo por instalación, mantenimiento y actualizaciones.

Para responder de manera efectiva a los usuarios, la Nube Computacional se divide en tres modelos de servicios: Software como Servicio (SaaS), Plataforma como Servicio (PaaS) e Infraestructura como Servicio (IaaS).

El SaaS como se describe en IBM (2014); Research (2013) y Cisco (2014) es uno de los servicios de la Nube Computacional más productivos y comúnmente utilizados hoy en día y se tiene pronosticado para los próximos años un crecimiento considerable, por el impacto en la economía y solución a problemas de negocios. Sin embargo, la complejidad actual del SaaS, por la necesidad de cubrir la creciente demanda ante un proceso dinámico en lo social, en la economía, tecnología y cultura de usuarios de la Nube, ha traído consigo desafíos importantes para asegurar que las aplicaciones desplegadas en el modelo de Software como Servicio cumplan satisfactoriamente las necesidades de los clientes.

La orquestación de servicios como se describe en Bousselmi *et al.* (2014): “*es uno de los principales elementos para explotar el potencial de la Nube Computacional*”. Específicamente, esta es una actividad primordial en la gestión de servicios que debería estar presente en las etapas de despliegue, aprovisionamiento y ejecución del SaaS, en un ambiente tan dinámico como el que presenta la Nube en el otorgamiento de servicios con la funcionalidad requerida. Sin embargo, algunas necesidades como: a) asegurar la correcta ejecución del flujo de trabajo del SaaS tomando en cuenta la coordinación de servicios, la asignación y coordinación de recursos heterogéneos en tiempo de ejecución y b) la eficiencia de la orquestación tomando en cuenta la sobrecarga de ejecución que esta pueda presentar en tiempo de ejecución, quedan aún sin ser del todo atendidas.

En este trabajo se presenta un modelo de orquestación dinámica de servicios atómicos de flujos de trabajo, orientado a atender el inciso a) antes mencionado. El modelo se implementa en el lenguaje de programación java y ubicado en el modelo PaaS. La implementación tiene el objetivo de probar la funcionalidad del modelo. El modelo se acopla al ambiente dinámico de la Nube Computacional para apoyar en la ejecución adecuada de los servicios atómicos que forman parte de un flujo de trabajo de la aplicación correspondiente a un proceso de negocios en el modelo de Software como Servicio.

El resto de este artículo está organizado de la siguiente manera: En la sección de literatura y trabajos relacionados se presenta una descripción del estado de la práctica sobre la composición y comportamiento del SaaS, así como la orquestación de servicios con sus principales desafíos, junto con un estudio en el campo de la orquestación de servicios con la intención de mostrar las soluciones propuestas más actuales. En la sección de Modelo de orquestación dinámica (MOD) se muestra este, así como las características que presenta para apoyar la ejecución exitosa de aplicaciones de procesos de negocios. En la sección de Evaluación del modelo se presentan las pruebas que permiten la evaluación de este, en donde se define el ambiente utilizado para probar el desempeño del modelo propuesto. En la sección de Análisis de resultados se presenta un análisis y discusión de los resultados del modelo de orquestación dinámica a partir de la solución del problema de disponibilidad que brinda al SaaS. Por último, se presentan las conclusiones y el trabajo futuro del modelo de orquestación dinámica, así como algunos trabajos que quedan por abordar.

## LITERATURA Y TRABAJOS RELACIONADOS

### COMPOSICIÓN Y COMPORTAMIENTO DE SAAS

En la actualidad la creciente demanda de los consumidores de software a la medida con requerimientos específicos de ambientes críticos, económicos y culturales, ha propiciado el desarrollo de un SaaS más complejo.

Para que el SaaS pueda brindar soluciones a sus clientes, este se despliega hoy en día en servicios a pequeña, mediana o gran escala. Tal como se describe en Fowler y Lewis (2014), los servicios a pequeña escala, también llamados servicios atómicos o microservicios, dan al SaaS la capacidad de poder brindar dinámicamente características de reusabilidad, escalabilidad, elasticidad y personalización de servicios en un esquema de flujos de trabajo que se apegue a procesos de negocios de los clientes.

En el SaaS se presentan tres conceptos importantes que vale la pena describir. El primero de ellos es la Composición de Servicios, que se refiere a la composición de servicios a partir de múltiples microservicios o paquetes de software expresados en cierta lógica en un flujo de trabajo (Wang *et al.*, 2014). Esta lógica obedece a algún proceso de negocios que atiende las necesidades específicas de ciertos usuarios. El segundo de ellos es el Flujo de Trabajo (*workflow*), que puede verse como un conjunto de pasos computacionales que requieren para su ejecución paquetes de terceros, los cuales pueden ser representados como componentes o funciones de software (Wang *et al.*, 2014). Estos paquetes comúnmente son llamados servicios atómicos o microservicios, como se describe en Fowler y Lewis (2014). Por último, la Automatización de Servicios se refiere a una de las tareas complejas en la gestión de los servicios, es decir, se refiere a la capacidad de los servicios en la Nube para adaptarse (elasticidad, extensibilidad) de forma automática y efectiva a las necesidades actuales de los usuarios, como se menciona en Klai y Ochi (2016).

Estos tres conceptos en conjunto dan pie a la necesidad de establecer servicios atómicos, o microservicios, con el grado de autosuficiencia necesaria para cumplir con una responsabilidad única, tener débil acoplamiento, aislamiento, alta disponibilidad, escalabilidad, reusabilidad, reactividad y alto rendimiento, así como autonomía, movilidad y estado exclusivo, características esenciales de los microservicios abordados en Bonér (2017).

Cada servicio de software, como se describe en Myerson (2009), puede actuar como un proveedor de servicios o como un solicitante de servicios. Un proveedor de SaaS expone las funcionalidades de sus servicios a otras aplicaciones a través de agentes públicos. Esto es para que el solicitante de servicios pueda incorporar a sus aplicaciones tanto datos como funcionalidades desde otros servicios. Ambos pueden aprovechar la economía por la escalabilidad en el despliegue y gestión del SaaS.

Para dar atención a los consumidores, el SaaS presenta el siguiente funcionamiento: Una aplicación (servicio) es ejecutada por un cliente en un entorno de Internet mediante un navegador Web. La interacción con el SaaS desencadena una secuencia de llamadas a otros servicios debido a que este se representa en un flujo de trabajo, el cual se definió previamente por el proveedor de servicios en el modelo PaaS, y está compuesto por un conjunto de servicios atómicos o microservicios, los cuales pueden ser del mismo proveedor de servicios o de terceros, normalmente alojados físicamente en lugares diferentes en la Nube y con recursos

físicos previamente asignados para su correcta ejecución. Esta interacción de servicios independientes, con diferentes arquitecturas y características, así como la dependencia que presentan con los recursos asignados a cada uno de ellos, pueden propiciar comportamientos no deseados en el desempeño del SaaS.

Los comportamientos anormales de SaaS son el principal desafío para proveedores e investigadores, puesto que impactan en la disponibilidad del servicio, la cual está relacionada directamente con el grado en el que un servicio puede ser utilizado. La disponibilidad es una de las principales características del modelo SaaS y uno de los principales objetivos a cumplir en el Acuerdo de Nivel de Servicios (Van-Der *et al.*, 2014).

El grado de disponibilidad que tiene un servicio para ser utilizado está relacionado con los recursos empleados por los microservicios que forman parte del flujo de trabajo de aplicaciones de procesos de negocios, la interoperabilidad entre estos, así como la capacidad que los mismos tengan para realizar la función para la que fueron creados.

Por ello, la búsqueda, selección, recomendación, coordinación de servicios virtualizados y no virtualizados, así como la coordinación y asignación dinámica de recursos necesarios para la correcta ejecución de servicios, son algunos de los desafíos a ser abordados por los investigadores del área.

#### ORQUESTACIÓN DE SERVICIOS Y SUS DESAFÍOS

La orquestación de servicios es uno de los principales elementos para explotar el potencial de la Nube Computacional. Dependiendo del lugar en donde se requiera llevar a cabo la orquestación, será la manera en cómo esta considerará y coordinará los servicios.

El término orquestación de servicios está muy ligado con el concepto de coreografía de servicios, sin embargo, existe una diferencia entre ambos conceptos. La coreografía de servicios, tal como se describe en Terpák *et al.* (2016) es una descripción global de la participación de servicios, la cual se define por el intercambio de mensajes, reglas de interacción y acuerdos entre dos o más servicios. Por su parte, la orquestación de servicios, como se describe en Terpák *et al.* (2016), representa el único proceso ejecutable centralizado. Este proceso ejecutable es el responsable de invocar y combinar los servicios.

De manera general, la orquestación de servicios se puede establecer en dos niveles como se describe en Bousselmi *et al.* (2014): La Orquestación de Software como Servicios (SaaSO) y la Orquestación de Infraestructura como Servicio (IaaSO). A continuación se defi-

ne SaaS por la relación directa con el tema de investigación abordado en el desarrollo tecnológico descrito en este artículo.

El SaaS consiste en establecer la lógica de negocios y el orden de ejecución de las tareas y puede abarcar la organización de las aplicaciones para definir un modelo transaccional del proceso de múltiples pasos a largo plazo. Puesto que los servicios son asociados con los recursos físicos donde estos son ejecutados, la orquestación de SaaS incluye la organización, coordinación y administración del despliegue y configuración automatizado de uno o más componentes interrelacionados, necesarios para el aprovisionamiento de servicios en un momento dado, por lo que es importante analizar una serie de técnicas de gestión de servicios y recuperación de fallas.

Como se menciona en Bousselmi *et al.* (2014), uno de los desafíos actuales en SaaS es la asignación dinámica de recursos, garantizando los Acuerdos de Nivel de Servicios (SLA) en los recursos de la Nube para tomar decisiones rentables en la colocación de la carga de trabajo y la coordinación dinámica de recursos, que permita optimizar el uso de los recursos de los servicios y, al mismo tiempo, reducir el tiempo de respuesta global de cada servicio.

Por otro lado, como se menciona en Weerasiri *et al.* (2017), en entornos de ejecución de servicios es necesario automatizar la orquestación que comprenda tanto el monitoreo como el control con la finalidad de asegurar la ejecución exitosa de los servicios, mejorando significativamente su productividad.

Existen algunos trabajos que presentan soluciones de orquestación de servicios en tiempo de despliegue para garantizar su calidad. En De Brito *et al.* (2017); Devanathan y Sundaramurthy (2016); Josa *et al.* (2016); Mayoral *et al.* (2017); Qi-Liao *et al.* (2016) y Soenen *et al.* (2017) se presenta una solución para asegurar el correcto funcionamiento de los servicios utilizados por el cliente. Por otro lado, Ranchal *et al.*, 2017; Luo & Salem (2016) garantizan la seguridad y privacidad entre las interacciones de los servicios orquestados y los clientes.

Para el caso de garantizar la calidad de servicios en tiempo de ejecución, en Bosman *et al.* (2015) se presenta una solución encaminada a la composición de servicios en tiempo real para su entrega al cliente.

En Ruiz *et al.* (2017) se ha presentado un modelo de orquestación dinámica aplicada al SaaS como una propuesta de solución encaminada a disminuir las fallas o comportamientos anormales de los servicios que participan en el proceso de ejecución de las aplicaciones de negocios (SaaS) en tiempo de ejecución. El modelo propuesto es sensible a factores dependientes e indepen-

dientes de los servicios atómicos que conforman la aplicación y propone, en tiempo de ejecución, la recomposición del flujo de trabajo con servicios que cuenten con el grado de disponibilidad necesaria para garantizar la ejecución exitosa del SaaS.

En este artículo se presenta un modelo de orquestación dinámica aplicada a flujos de trabajo del SaaS, el cual se ubica en el modelo de PaaS. Este modelo es una evolución y maduración del modelo propuesto en Ruiz *et al.* (2017), logrando obtener un modelo completamente funcional en términos de orquestación en tiempo de ejecución de los servicios atómicos que forman parte del flujo de trabajo del SaaS. El modelo en esta versión solo contempla la orquestación dinámica a partir de la revisión en tiempo de ejecución de factores dependientes e independientes de los servicios atómicos que cumplen con la disponibilidad y confiabilidad necesaria para su correcta ejecución, acoplándose con ello al ambiente dinámico que presenta la Nube para la ejecución en tiempo y forma del SaaS en el momento de presentar sus servicios al cliente.

## MODELO DE ORQUESTACIÓN DINÁMICA

Dado los desafíos presentados en SaaS, donde el servicio pudiera no cumplir en tiempo y forma o no llegara a producir los resultados esperados al momento de ser utilizado por un cliente, debido al grado de disponibilidad que presentan los servicios atómicos por la presencia de diferentes factores que afectan su desempeño, en este artículo se presenta un Modelo de Orquestación Dinámica (MOD) de servicios atómicos de flujos de trabajo de aplicaciones de negocios, con el objetivo de apoyar la ejecución exitosa de los servicios participantes en flujos de trabajo de procesos de negocios.

El MOD de servicios es sensible a factores de comportamiento que presentan los servicios. Este modelo contempla en su proceso monitorear y recolectar, en tiempo de ejecución, la información correspondiente a las características de comportamiento presentadas en los servicios atómicos del flujo de trabajo asociados a servicios de procesos de negocios de clientes, como son: a) factores dependientes (que se mantienen de manera estática) de los servicios: costo, tiempo de procesamiento, capacidad de respuesta por llamadas al servicio, reputación; b) factores independientes (que varían dinámicamente) de los servicios tales como: ancho de banda, latencia de servidores, rendimiento de un servicio sobrecargado (*throughput*), disponibilidad de los servicios, número de peticiones re-entrantes de servicios.

La Figura 1 ilustra el proceso de orquestación dinámica de servicios atómicos, de un flujo de trabajo co-

rrespondiente a un SaaS, que efectúa el modelo propuesto.

El módulo de Solicitud del Workflow del proceso de orquestación dinámica del modelo toma como entrada el archivo de composición correspondiente a un flujo de trabajo proveniente de la aplicación de un proceso de negocios. Este archivo es analizado para establecer el tipo de flujo de trabajo (sintético o proceso ejecutable). Una vez identificado, el flujo de trabajo se envía al módulo de Análisis del Workflow, que es el encargado de identificar cada uno de los servicios atómicos que lo conforman, así como la profundidad en el flujo y los recursos asignados para su ejecución. Cada servicio atómico del flujo de trabajo se analiza por el Coreógrafo para determinar los factores descritos en a) y b) del párrafo anterior. El resultado de este análisis determina servicios atómicos con factores a) y b) no deseables. El Coreógrafo, a su vez, interactúa con dos sistemas de recomendación de servicios sustitutos. Estos sistemas se utilizan para encontrar servicios funcionalmente equivalentes a los servicios no deseables, los evalúa en sus factores a) y b), y presenta una lista ordenada de servicios alternativos al Orquestador. El proceso del modelo de orquestación, determina a partir de su módulo Orquestador los servicios candidatos, la carga de trabajo de los recursos disponibles, los recursos asignados a cada servicio, así como el desempeño de los servicios a partir de los recursos asignados para determinar la probabilidad de falla de

cada uno de ellos, con el fin de seleccionar los servicios sustitutos que mejor ajusten al actual flujo de trabajo de la aplicación del cliente.

Una vez establecidos los servicios atómicos que participarán en el flujo de trabajo, el modelo gestiona, en su módulo Re-composición del Workflow, la secuencia ordenada que integra el flujo de control del archivo de composición de servicios para establecer una nueva versión del flujo de trabajo de la aplicación del proceso de negocios del cliente. Este proceso está integrado con los servicios atómicos, que al momento tienen las mejores condiciones de disponibilidad en tiempo de ejecución.

Una parte importante del MOD es que este se encuentra desplegado como servicio en el modelo PaaS, como se muestra en la Figura 1. Esta posición permite que, a partir de las plataformas utilizadas por el proveedor de servicios, se ejecute como servicio en segundo plano (demonio) permitiendo con ello tener acceso a la parte subyacente de la instancia (contenedor), en donde se sitúa el SaaS, para interactuar con el planificador de tareas, con la finalidad de descubrir los recursos asignados a cada servicio, así como el archivo de composición para identificar los servicios atómicos o microservicios, que participan en el flujo de trabajo; monitorear sus factores dependientes e independientes con el fin de establecer en tiempo de ejecución el grado de disponibilidad y confiabilidad para la ejecución en tiempo y forma del SaaS al que representan.

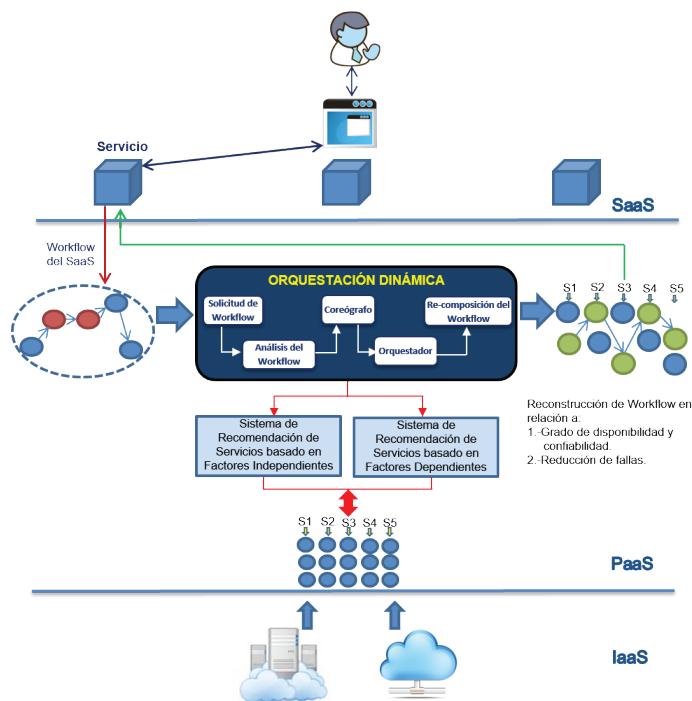


Figura 1. Modelo de orquestación dinámica de servicios atómicos

Este MOD soporta la característica multi-alquiler (multitenancy), ya que al convivir con las plataformas de despliegue del SaaS permite analizar el dominio utilizado, identificando con ello al cliente que accede al SaaS desde una instancia. Sin embargo, por el momento, el MOD no analiza la funcionalidad a partir de las configuraciones establecidas a nivel de interfaz en tiempo de ejecución, solo analiza las configuraciones establecidas para cada instancia de cliente por medio del planificador de tareas.

### EVALUACIÓN DEL MODELO

Para efectos de evaluación del modelo, este tuvo que implementarse en un software, el cual se puso en funcionamiento como un componente de software en un ambiente de Nube Computacional simulada. Esta Nube fue creada utilizando la herramienta WorkflowSim (Chen y Deelman, 2012).

El componente MOD se instancia e inicia su ejecución durante la iniciación de los planificadores, contenedores, servicios utilizados (cloudlets) y el motor de flujos de trabajos en el modelo PaaS durante la ejecución de la herramienta WorkflowSim, como se muestra en el proceso de la Figura 2.

El MOD fue desarrollado bajo un esquema de plataforma de propósito general en donde se contempla una diversidad de lenguajes de desarrollo, así como mecanismos de escalabilidad automática, contenedores y middleware estandarizados para el despliegue de aplicaciones SaaS.

En el ambiente simulado se creó un centro de datos con dos servidores, cada uno con las siguientes características: dos procesadores de doble núcleo de 113093 mips; 16GB en RAM; 1 TB de almacenamiento; Sistema Operativo Windows y un monitor de máquinas virtuales Xen. Cada servidor contiene 20 máquinas virtuales, cada una con las siguientes características: un procesador de 1000 mips, 512 en RAM, ancho de banda de 1000 mps y 10 GB de almacenamiento. Con un gestor actuando como intermediario entre el cliente y el host.

Se utilizó como caso de prueba el flujo de trabajo sintético de un archivo DAX (Directed Acyclic Graph in XML), con la sintaxis para expresar trabajos, argumentos, archivos y dependencias; del servicio Montage (Pegasus, 2010), con 100 servicios atómicos, representando cada uno de ellos un trabajo diferente para dar atención a la aplicación Montage ejecutada en el ambiente de Nube Computacional simulado.

Para la evaluación se ejecutaron llamadas al servicio Montage con una tasa de sobrecarga al ambiente simulado de 10% con el fin de provocar algún porcentaje de fallas en los servicios atómicos y los recursos asignados a cada uno de ellos, asimismo para verificar la capacidad del modelo y encontrar fallas o servicios anómalos, aún cuando exista una tasa baja de estos.

Las funcionalidades que se probaron del modelo de orquestación dinámica fueron: la identificación de servicios atómicos en el flujo de trabajo de entrada, la evaluación de servicios atómicos en tiempo de ejecución, a partir de los valores correspondientes a sus atributos dependientes e independientes y la ejecución del flujo de trabajo reconstruido para determinar su resultado.

El MOD inicia en una instancia independiente de los servicios atómicos, o microservicios, utilizados por el consumidor, evitando sobrecargar los recursos de los servicios en ejecución. Debido a que el Motor de Workflow de la herramienta WorkflowSim se basa en eventos, al iniciar la ejecución de un SaaS el MOD identifica su inicio y recibe su archivo de composición permitiendo identificar los servicios atómicos correspondientes a su flujo de trabajo. Con estos datos el MOD recaba información de los servicios a partir de los módulos de recomendación basados en factores dependientes e independientes. Una vez establecidos los servicios atómicos que serán sustituidos, el MOD, a través del planificador, enviará los servicios sustitutos a la cola de tareas sustituyendo aquellos que fueron pronosticados como fallidos por los servicios sustitutos para su ejecución. Como último paso, el MOD reescribirá el archivo de composición con los servicios sustitutos seleccionados.

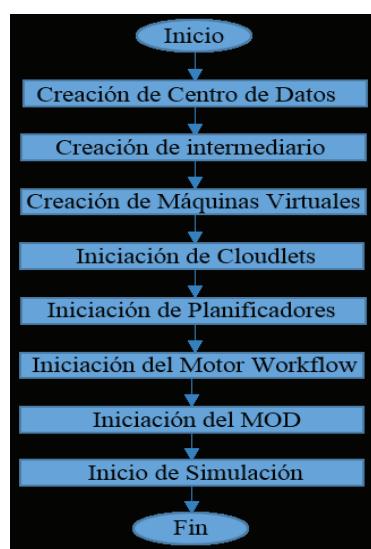


Figura 2. Proceso de inicio de la simulación utilizando MOD

## ANÁLISIS DE RESULTADOS

A partir de la entrada del flujo de trabajo sintético correspondiente al servicio de Montaje, la implementación del modelo de orquestación dinámica identificó aquellos servicios atómicos presentes en el flujo de trabajo, así como la profundidad de estos en este flujo de trabajo; asimismo, se identificaron los recursos asignados y el número de tareas establecidas para cada uno, tal como se muestra en la Figura 3.

Con esta información es posible revisar los valores asociados a los atributos tanto dependientes como independientes que en un momento dado determinan la disponibilidad de los servicios atómicos en el flujo de trabajo.

La evaluación consistió en medir la distancia existente entre la cantidad de servicios atómicos obtenidos mediante la herramienta WorkflowSim y la cantidad de servicios atómicos obtenidos mediante el uso del modelo de orquestación dinámica colocado en el ambiente simulado. Para esta evaluación se realizaron 10 ejecuciones del flujo de trabajo con una tasa de generación de fallos de 10% utilizando la herramienta WorkflowSim sin el uso del MOD, obteniendo el comportamiento final de 2000 servicios atómicos, donde fueron identifi-

cados tanto servicios fallidos como aceptados (ejecutados correctamente). Para contrastar el resultado, se realizaron otras 10 ejecuciones con las mismas condiciones que la prueba anterior, pero ahora conectando el MOD a la ejecución del WorkflowSim. En esta prueba se identificó, en tiempo de ejecución, el comportamiento de 2000 servicios atómicos en donde se encontraron tanto servicios con condiciones de falla como aceptados (con condiciones de ejecución adecuada).

La precisión de esta distancia se mide en el rango de 0 a 1, donde 0 significa baja precisión o una gran distancia en el desempeño comparativo de ambas herramientas. Al contrario, 1 significa alta precisión o una distancia de 0 en el desempeño de ambas herramientas.

El índice de precisión se determinó a partir de la siguiente métrica:

$$P = \frac{TSMOD}{TSS} \quad (1)$$

Donde TSMOD es el total de servicios atómicos identificados por el modelo de orquestación dinámica en la ejecución de la simulación y TSS es el total de servicios atómicos generados como salida por el simulador.

La Tabla 1 muestra que para el caso de prueba de la funcionalidad del modelo de orquestación dinámica

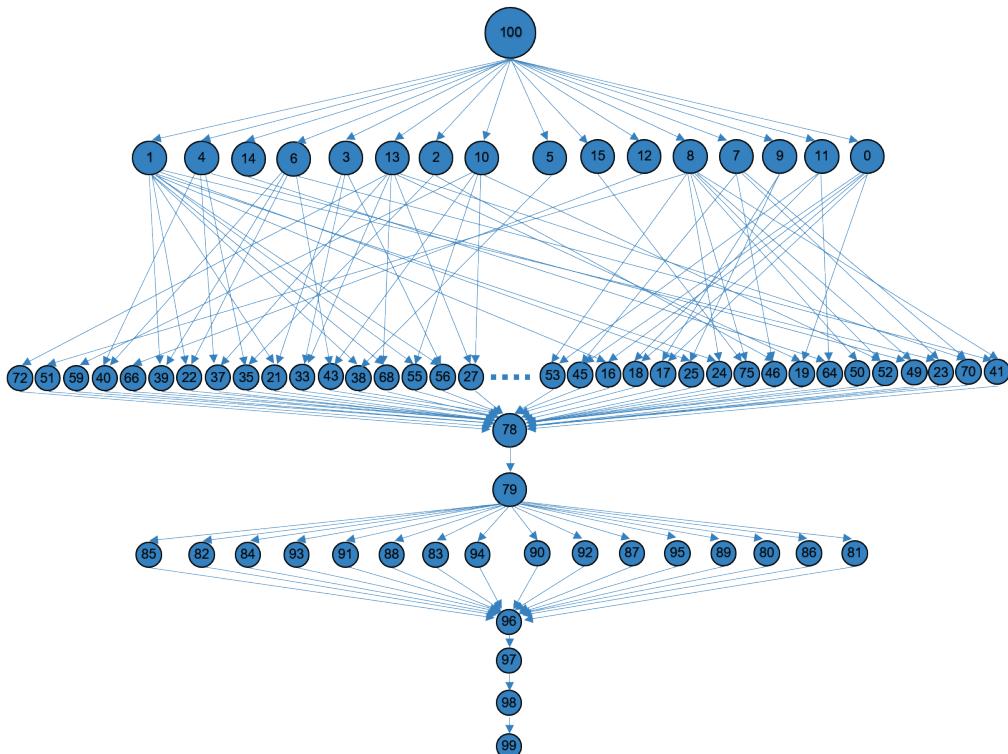


Figura 3. Servicios atómicos involucrados en el flujo de trabajo

basada en la revisión de valores asociados a los factores dependientes e independientes, se obtuvo un índice de precisión igual a 1 en la identificación del grado de disponibilidad con que cuentan los servicios atómicos del flujo de trabajo. Este grado de disponibilidad etiqueta a los servicios como aceptados o fallidos.

La precisión de los resultados emitidos por MOD puede observarse gráficamente en la Figura 4, en donde se presentan cada uno de los resultados emitidos por las 10 ejecuciones del simulador con una tasa de generación de fallos de 10%. Se observa que los hallazgos obtenidos al utilizar el MOD, en tiempo de ejecución, corresponden con los resultados emitidos por el simulador, en relación con los servicios atómicos que fueron ejecutados exitosamente y aquellos que presentaron fallas en su ejecución.

Para el caso de los servicios atómicos etiquetados como fallidos por el modelo de orquestación, debido a las características que presentaron en sus factores dependientes e independientes, el modelo de orquestación definió tres servicios atómicos alternativos por cada servicio fallido, definiendo un total de 489 servicios atómicos alternativos de los cuales 163 son los sustitutos a ser ejecutados.

Para determinar la probabilidad de falla, el modelo de orquestación contempla la probabilidad de espacios

equiprobables presentada en Seymour & Marc (2009) y definida en la ecuación 2 para valores obtenidos en tiempo de ejecución del comportamiento que presentan los servicios atómicos.

$$P(E) = \frac{\text{número de resultados desfavorables a } E}{\text{número de resultados posibles de } S} \quad (2)$$

Donde  $S = \{A \cup B \cup C \cup D\}$  es el espacio muestral y  $E = \{A \cap B \cap C \cap D\}$  es el evento que constituye un conjunto de resultados con los servicios atómicos cuyas características presentadas obedecen al conjunto de servicios atómicos  $S_{PF} = \{x \in S_{bp} | x \in A, x \in B, x \in C, x \in D\}$  que representa anomalías al SaaS (probabilidad de fallas). Donde  $A = \{x \in S_{bp} \cap x \in N_{cf}\}$ ,  $B = \{x \in S_{bp} \cap x \in N_{tp}\}$ ,  $C = \{x \in S_{bp} \cap x \in N_d\}$ ,  $D = \{x \in S_{bp} \cap x \in N_{clo}\}$ . Donde  $N_{cf}$  es el grado de confiabilidad  $\leq 80\%$ ,  $N_{tp}$  es tiempo de respuesta  $\leq 80\%$ ,  $N_d$  es el grado de disponibilidad  $\leq 80\%$  y  $N_{clo}$  es el costo comprendido en el rango de  $+/- 10\%$  del costo original.

A continuación en la Tabla 2 se muestra el resultado de servicios atómicos al utilizar el MOD en el ambiente simulado. De las 10 ejecuciones realizadas utilizando solo la herramienta WorkflowSim se puede visualizar que de los 163 fallos no se realizan propuestas de servi-

Tabla 1. Índice de precisión en la revisión dinámica de servicios atómicos

WorkflowSim sin MOD		WorkflowSim con MOD	
Servicios	2000	Servicios	2000
Aceptados	1837	Aceptados	1837
Fallidos	163	Fallidos	163

Caso para servicios atómicos aceptados	P= 1837/1837=1
Caso para servicios atómicos fallidos	P= 163/163=1

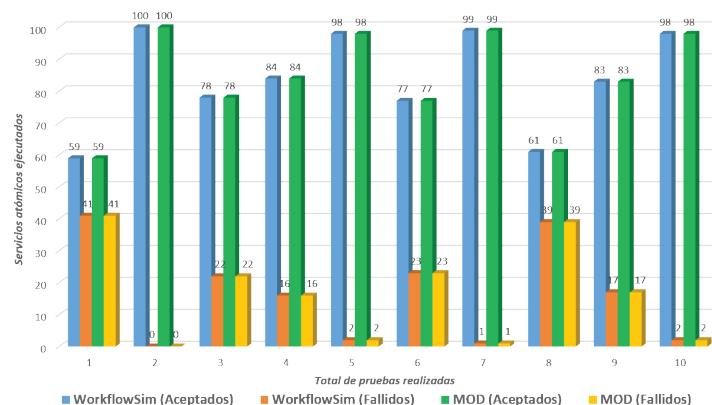


Figura 4. Identificación del comportamiento de servicios atómicos

cios alternativos ni sustitutos en tiempo de ejecución para evitar la falla de los servicios. Al final de cada una de las ejecuciones la herramienta marca a cada uno de los servicios como fallidos dando un total de 1837 servicios ejecutados como aceptados, para las 10 ejecuciones. Para el caso de las 10 ejecuciones utilizando el MOD como orquestador de la herramienta WorkflowSim para la ejecución exitosa de la aplicación, se puede observar que el MOD también es capaz de identificar en tiempo de ejecución los 163 servicios con comportamientos de fallas, proponiendo tres servicios con características deseables para una correcta ejecución, y para cada servicio identificado como fallido dando un total de 489 servicios atómicos alternativos para evitar las fallas. De los servicios alternativos se seleccionaron 163 servicios sustitutos a ser colocados en el flujo de trabajo alternativo. Lo anterior, a partir del grado de satisfac-

ción de los requerimientos de disponibilidad y carga de trabajo en los recursos asignados a los servicios. El flujo de trabajo alternativo se envía en tiempo de ejecución a la herramienta WorkflowSim para planificar los servicios sustitutos y se ejecutan en el ambiente simulado. En total para las 10 ejecuciones de flujo de trabajo de Montaje utilizando WorkflowSim con el MOD, se ejecutaron 2000 servicios atómicos exitosamente.

Para mayor detalle, en la Figura 5 se puede observar el desempeño del MOD en la selección y ejecución de servicios atómicos candidatos en cada una de las 10 pruebas realizadas en el ambiente simulado.

Los servicios atómicos alternativos (representados por la barra en color azul), los cuales se obtuvieron de la selección de todos los servicios sustitutos (representados por la barra amarilla con azul), sustituyen a todos aquellos servicios atómicos que presentaron condicio-

Tabla 2. Resultados de servicios atómicos al utilizar mod

Servicios encontrados	WorkflowSim	Aceptados	1837
		Fallidos	163
Servicios alternos	WorkflowSim con MOD	Aceptados	1837
		Fallidos	163
Servicios candidatos	WorkflowSim	Alternativos	0
	WorkflowSim con MOD	Alternativos	489
Servicios ejecutados exitosamente	WorkflowSim	Sustitutos	0
	WorkflowSim con MOD	Sustitutos	163

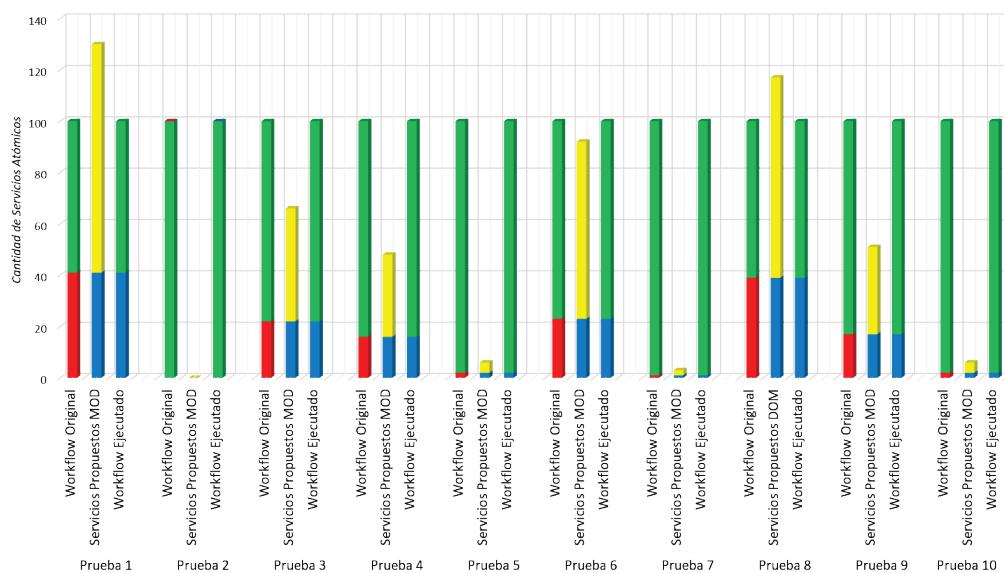


Figura 5. Desempeño del MOD en la ejecución de servicios atómicos

nes de falla (representados por la barra en color rojo) en el flujo de trabajo Montage original. El flujo de trabajo reconstruido (representado por la barra de color verde con azul) fue ejecutado exitosamente en el ambiente simulado generado por WorkflowSim.

### CONCLUSIONES Y TRABAJO FUTURO

Debido al ambiente dinámico que presenta la Nube Computacional es difícil anticipar los cambios en el ambiente que puedan afectar el estado del SaaS poniendo en peligro su desempeño.

En el SaaS un desafío que requiere ser abordado en su totalidad es la disponibilidad de los servicios, que integran el flujo de trabajo, para garantizar el funcionamiento correcto de la aplicación del consumidor del servicio.

Hasta el momento, no existe en la literatura un modelo de orquestación dinámico encaminado a controlar el rendimiento del SaaS en tiempo de ejecución a partir de la disponibilidad real de los servicios atómicos que lo conforman. El modelo de orquestación dinámica de servicios presentado en este artículo contribuye de manera importante en el control del rendimiento del SaaS, ya que permite emplear una técnica proactiva que se anticipa a fallos, eligiendo aquellos servicios atómicos que serán remplazados y aquellos que los remplazarán mediante el análisis, en tiempo de ejecución, de los atributos correspondientes a los factores dependientes e independientes que se presentan en los servicios. La gestión de servicios considera la asignación de recursos mediante el empleo de técnicas de cargas de trabajo y gestión de flujos de trabajo para garantizar su correcta ejecución.

Para la coordinación efectiva de los servicios atómicos del flujo de trabajo, el modelo de orquestación dinámica de servicios considera el monitoreo y coordinación de servicios atómicos, así como los factores de influencia dependientes e independientes.

Los resultados obtenidos del modelo de orquestación dinámica en el ambiente de Nube Computacional simulada, demuestran que el MOD determina las condiciones favorables o desfavorables de ejecución de los servicios atómicos en tiempo de ejecución, con base en la revisión dinámica de los valores presentados en los atributos dependientes e independientes de servicios atómicos del flujo de trabajo de la aplicación Montage, antes que estos sean utilizados en el flujo de trabajo, permitiendo aplicar, en caso de ser necesario, la reconstrucción de este mediante la sustitución de servicios con baja disponibilidad en el flujo de trabajo original,

por servicios atómicos alternativos con la disponibilidad necesaria para una correcta ejecución del SaaS.

Una aportación importante del MOD es la definición, en tiempo de ejecución, de un conjunto de servicios alternativos, como se describió en el capítulo de Análisis de resultados. Esta aportación permite la generación de flujos de trabajos alternativos ordenados de mayor a menor, con base en criterios de disponibilidad para asegurar la correcta ejecución del SaaS.

Se pudo determinar que los valores asociados a los servicios atómicos tales como:

- Factores dependientes indicados en el inciso a) de la sección 3 (estáticos): costo, tiempo de procesamiento, capacidad de respuesta, reputación.
- Factores independientes indicados en el inciso b) de la sección 3 (dinámicos): ancho de banda; latencia de servidores; rendimiento de un servicio sobre cargado (throughput); disponibilidad de servicio; número de peticiones re-entrantes.

Estos permiten determinar en tiempo de ejecución la disponibilidad real con la que cuentan los servicios atómicos de un flujo de trabajo correspondiente a un SaaS de la Nube Computacional.

Aunque la eficiencia no fue un objetivo perseguido en este trabajo de investigación, las técnicas empleadas de planificación, recuperación, monitoreo y recomendación utilizados, fueron dirigidas a no generar sobre carga en la ejecución del modelo.

Algunos de los trabajos que quedan por abordar en el futuro para la orquestación de servicios en un ambiente dinámico son:

- Revisión dinámica a partir de un histórico de comportamiento que presenten los servicios atómicos a lo largo de su uso, con la intención de reducir el desgaste en las revisiones dinámicas de los factores dependientes e independientes de los servicios atómicos.
- El control de versiones de los servicios atómicos con presencia de anomalías en tiempo de ejecución.
- Mejorar la eficiencia del modelo propuesto de orquestación dinámica, generando un método de orquestación ligero que haga uso adecuado de los recursos en tiempo de ejecución.
- Una orquestación dinámica de servicios que actúe automáticamente durante el despliegue, aprovisionamiento y ejecución de servicios atómicos.
- Realización de la orquestación de servicios que considere todas las condiciones establecidas entre el cliente y proveedor en el SLA.

- Orquestación dinámica de servicios que considera la migración de instancias virtuales, máquinas virtuales y máquinas físicas que alojen servicios atómicos a ser coordinados durante el despliegue, aprovisionamiento y ejecución del mismo.

### AGRADECIMIENTOS

Al Tecnológico Nacional de México/Centro Nacional de Investigación y Desarrollo Tecnológico (TecNM/CENIDET) por el apoyo brindado para la realización de la presente investigación en el programa de doctorado, así como por el financiamiento al proyecto de investigación No. 5517.15.16-PR. Al CONACyT por el apoyo de beca de doctorado que hizo posible la realización de esta investigación.

### REFERENCIAS

- Bonér, J. (2017). *Reactive microsystems. The evolution of microservices at scale*. En Foster B., editor. First ed., Sebastopol, CA: O'Reilly Media.
- Bosman, J., Van Den-Berg, H., Van Der-Mei, R. (2015). Real-Time QoS control for service orchestration. En 27th International Teletraffic Congress, 152-158. Recuperado de <https://doi.org/10.1109/ITC.2015.25>
- Bousselmi, K., Brahmi, Z., Gammoudi, M.M. (2014). Cloud services orchestration: A comparative study of existing approaches. En 28th International Conference on Advanced Information Networking and Applications Workshops, 410-416. Recuperado de <https://doi.org/10.1109/WAINA.2014.72>
- Chen, W. & Deelman, E. (2012). WorkflowSim: A toolkit for simulating scientific workflows in distributed environments. En IEEE 8th International Conference on E-Science, 1-8. Recuperado de <https://doi.org/10.1109/eScience.2012.6404430>
- Cisco. (2014). Cisco Global Cloud Index : Forecast and Methodology, 2014-2019. White Paper, 1-41. Recuperado de <https://doi.org/146130365180644>
- De Brito, M.S., Hoque, S., Magedanz, T., Steinke, R., Willner, A., Nehls, D., Schreiner, F. (2017). A service orchestration architecture for Fog-enabled infrastructures. En 2nd International Conference on Fog and Mobile Edge Computing, FMEC 2017, 127-132. Recuperado de <https://doi.org/10.1109/FMEC.2017.7946419>
- Devanathan, V. & Sundaramurthy, S. (2016). A Novel method and environment for scalable web services orchestration. En Proceedings IEEE World Congress on Services, SERVICES 2016, 128-129. Recuperado de <https://doi.org/10.1109/SERVICES.2016.27>
- Fernando, R., Ranchal, R., Bhargava, B., Angin, P. (2017). A monitoring approach for policy enforcement in cloud services. En IEEE 10th International Conference on Cloud Computing (CLOUD), 600-607. Recuperado de <https://doi.org/10.1109/CLOUD.2017.82>
- Fowler, M. & Lewis, J. (2014). Microservices. Recuperado de <https://martinfowler.com/articles/microservices.html>
- IBM. (2014). Champions of software as a service: How SaaS is fueling powerful competitive advantage, 11. Recuperado de <http://www.jsaf.com/common-examples-of-software-as-a-service.php>
- Josa, J.M.G., Lopez, V., Slyne, F., Ruffini, M., Vilalta, R., Mayoral, A., Martínez, R. (2016). End-to-end service orchestration from access to backbone. En 20th International Conference on Optical Network Design and Modeling, ONDM, 9(6), 137-147. Recuperado de <https://doi.org/10.1109/ONDM.2016.7494064>
- Klai, K. & Ochi, H. (2016). Model checking of composite cloud services. En IEEE International Conference on Web Services (ICWS), 356-363. Recuperado de <https://doi.org/10.1109/ICWS.2016.53>
- Luo, S. & Salem, M.B. (2016). Orchestration of software-defined security services. En IEEE International Conference on Communications Workshops, ICC, 436-441. Recuperado de <https://doi.org/10.1109/ICCW.2016.7503826>
- Mayoral, A., Vilalta, R., Muñoz, R., Casellas, R., Martínez, R., Svaluto-Moreolo, M., Kitayama, K. (2017). Control orchestration protocol: Unified transport API for distributed cloud and network orchestration. *Journal of Optical Communications and Networking*, 9(2), A216. <https://doi.org/10.1364/JOCN.9.00A216>
- Myerson, J.M. (2009). The role of software as a service in cloud computing SaaS maturity, abril, 1-8.
- Pegasus. (2010). Montage characterization-Pegasus workflow management system. Recuperado de <https://confluence.pegasus.isi.edu/display/pegasus/Montage+Characterization>
- Qi, Q., Liao, J., Wang, J., Li, Q., Cao, Y. (2016). Dynamic resource orchestration for multi-task application in heterogeneous mobile cloud computing. En Proceedings-IEEE INFOCOM, Septiembre (1), 221-226. Recuperado de <https://doi.org/10.1109/INFCOMW.2016.75620>
- Research, N. (2013). *El estado actual de cloud computing en México*. México: InformationWeek.
- Ruiz-Paz, S.F., Santaolaya-Salgado, R., Fragoso-Díaz, O.G. (2017). Model of dynamic orchestration for SaaS. *Revista Ingenierías Universidad de Medellín*, 16(31), 143-153. <https://doi.org/10.22395/rium.v16n31a7>
- Seymour, L. & Marc, L. (2009). *Matemáticas discretas*, 3a. ed. México: Mc Graw Hill.
- Soenen, T., Tavernier, W., Colle, D., Pickavet, M. (2017). Optimizing microservice-based reliable NFV management & orchestration architectures. En Proceedings of 2017 9th International Workshop on Resilient Networks Design and Modeling, RNDM, 1-7. Recuperado de <https://doi.org/10.1109/RNDM.2017.8093034>
- Terpák, J., Horovčák, P., Lukáč, M. (2016). Mathematical models creation using orchestration and choreography of web servi-

- ces. En 17th International Carpathian Control Conference (ICCC), 739-742. Recuperado de <https://doi.org/10.1109/CarpathianCC.2016.750>
- Van Der Wees A., Daniele, C., Jesus, L., Edwards, M., Schifano, N., Maddalena, S.L. (2014). Cloud service level agreement standardisation guidelines. *C-Sig Sla*, 1-41. <https://doi.org/10.1002/9781118691779>
- Wang, J., Korambath, P., Altintas, I., Davis, J., Crawl, D. (2014). Workflow as a service in the cloud: Architecture and scheduling algorithms. *Procedia computer science*, 29 (Supplement C), 546-556. <https://doi.org/10.1016/j.procs.2014.05.049>
- Weerasiri, D., Barukh, M.C., Benatallah, B., Sheng, Q.Z., Ranjan, R. (2017). A taxonomy and survey of cloud resource orchestration techniques. *ACM Computing Surveys*, 50(2), 1-1. <https://doi.org/10.1145/3054177>