

# CHOReOS Middleware

Andrés Merlo Trujillo

Universidad de Granada (UGR)

**Abstract.** En Internet cada vez se hacen más uso de servicios y de tecnologías distribuidas. Por tanto, surge la necesidad de intentar interconectar varios servicios, con características muy distintas para que puedan trabajar al unísono, creando así más servicios. Además, debido al uso masivo de servicios y el gran tráfico que tienen, se hace también necesario que estos servicios tengan garantías de disponibilidad y escalabilidad. Es por eso que ha surgido CHOReOS, para poder realizar todas estas tareas anteriormente mencionadas de una manera rápida y más sencilla que como tradicionalmente se hace. En este documento se describe el middleware, sus características, componentes y algún ejemplo de funcionamiento en la vida real.

**Keywords:** Servicios · Coreografías · Middleware.

## 1 Introducción

El aumento de tamaño de internet y el uso de web services ha destapado graves problemas para la incorporación de servicios en sistemas web. Este problema se puede resolver mediante estándares, haciendo uso de la orquestación de servicios. Sin embargo, conforme ha ido creciendo, se han descubierto graves problemas de escalabilidad y puntos críticos de fallo, debido a la naturaleza centralizada de la orquestación. [1]

Al haber crecido tanto internet, se ha convertido en una red heterogénea, móvil y adaptable donde tiene más sentido reutilizar servicios existentes de manera distribuida y descentralizada para los programas, en vez de realizar programas monolíticos [2].

Esta solución resolvería la centralización y los problemas anteriormente mencionados, con la desventaja de ser más difíciles de gestionar por su naturaleza distribuida [1].

A este concepto de reutilizar servicios de manera distribuida se le denomina “Coreografías”. Una coreografía permite realizar servicios distribuidos, formalizando la forma en la que los demás servicios interactúan, indicando el comportamiento esperado de los servicios participantes.

Es por eso que aparece CHOReOS, un middleware para componer coreografías de servicios a gran escala orientado principalmente al Internet del Futuro (FI) [1].

## 2 Objetivos

El objetivo principal de CHOReOS es el de ofrecer una plataforma para diseñar, desplegar y ejecutar las coreografías para sistemas orientados a servicios (SOS) y a gran escala. Para ello, CHOReOS ofrece un IDRE (Integrated Development and Runtime Environment) para modelar y especificar estas coreografías. Además tiene herramientas de validación que permiten verificar que la especificación realizada tenga un mínimo de calidad. [2]

Otro objetivo muy importante es el de mantener la escalabilidad, la interoperabilidad y el QoS (Quality of Service) [3]. Esto es algo también muy importante, ya que tienen pensado su uso para el Internet Futuro (FI) de ultra gran escala (ULS) [2].

Por último, otro objetivo que se puede deducir de los anteriores, es que han querido crear una capa de abstracción sobre la que trabajar para facilitar la creación de coreografías con unos mínimos de calidad. Estos mínimos se comprueban mediante el IDRE en la etapa de validación y verificación, si la calidad no es satisfactoria, se vuelve a la etapa inicial de especificación [2].

## 3 Diseño

El diseño de la arquitectura que proponen, se basan en otras tecnologías middleware: bus de servicio distribuido (DSB) y grid y cloud computing. [1]

Para la parte del DSB, hacen uso de una solución middleware denominada “PEtALS”, la cual se basa en una red Peer to Peer (P2P). Esto permitirá hacer un DSB muy escalable, que permitirá coreografías de servicios muy heterogéneos, muy indicado para el FI (Future Internet) [1].

El DSB se encarga principalmente de la conexión con servicios distribuidos de distintas organizaciones. Además, también se usará para el IDRE para poder descubrir los servicios distribuidos y poder definir la interacción con estos. [4]

Otro aspecto de diseño es el uso de una arquitectura de grid y cloud computing, que permite tener una escalabilidad y un rendimiento excelente.

Además, todos los servicios que estén disponibles para las coreografías, dispondrán de un adaptador para solventar las diferencias de comunicación y, a su vez, de un “Coordinator Delegate” (CD), que se encargará de realizar la coordinación entre los servicios elegidos para cada coreografía. Estas entidades se intercambian la información de coordinación para evitar interacciones incorrectas. [2]

## 4 Características

Las principales características que tendrá este middleware son las siguientes:

- **Rendimiento:** Gracias al uso del Grid y Cloud Computing, se pueden servir a millones de usuarios, realizando miles de peticiones simultáneas. [1]
- **QoS-aware:** Haciendo uso de un subcomponente denominado “Predictor”, el middleware puede estimar comportamiento de la calidad de servicio frente al tiempo [4], haciendo que otros subcomponentes puedan tomar decisiones.

- **Escalabilidad:** La arquitectura especificada permite que se puedan lanzar muchos servicios sin repercutir demasiado en el rendimiento. Además, haciendo uso del “Predictor” y un analizador de escalabilidad (Scalability Analyser), se puede estimar la evolución del QoS, algo directamente relacionado con la escalabilidad. [4]
- **Tolerante a fallos:** Debido a la naturaleza distribuida de los servicios, estos pueden estar replicados y poder continuar el funcionamiento del sistema sin problema. Además, el middleware posee un mecanismo de tolerancia a fallos [5], que se encarga de mantener el servicio siempre activo.
- **Simplicidad en la modelación de coreografías:** Gracias al uso del IDRE, se puede modelar coreografías de servicios de manera sencilla, permitiendo especificar los requisitos y modelar las coreografías. La coreografía es creada automáticamente, asegurando los criterios de calidad y escalabilidad [2].

## 5 Mecanismos

El principal mecanismo para la coordinación de los servicios es mediante el uso de paso de mensajes en los CD. Los CD se encargan de enviar información para asegurarse en todo momento de que se estén realizando las coreografías correctamente.

Los mensajes propios de los servicios para realizar la tarea deseada en un sistema web, se intercambian siguiendo el protocolo petición/respuesta (Request/Response RR). Además, estos mensajes deben ser enviados por los propios CD [2], luego pasar por el adaptador del servicio, para traducirlo y que sea entendible por el mismo, y luego realizar la operación deseada.

Adicionalmente, en caso de caída de un servicio, el middleware debe ser capaz de reasignar a las coreografías que usen dicho servicio a otra réplica, siendo transparente para los clientes.

## 6 Protocolos y servicios

Para la interconexión de las distintas componentes de CHOReOS, se ha usado una API REST sobre HTTP, ya que es superior en términos de rendimiento, velocidad y fiabilidad que otros protocolos. [6]

Esto permite tener una interfaz uniforme entre todos los componentes, haciendo que la comunicación y la depuración sea más sencilla frente a otras opciones como DPWS. [6]

Este middleware consta principalmente de un servicio, denominado “Enactment Engine” (EE). Este se encarga de manejar los fallos de los componentes externos, muy importante en sistemas a gran escala [6].

Además, este servicio consta de los siguientes componentes:

- **Choreography Deployer:** Expone la API para dar soporte al despliegue de coreografías. El cliente debe suministrar la especificación de la coreografía,

esta contiene la localización de los servicios [6] y la forma de interacción de los mismos.

- **Deployment Manager:** Se encarga de desplegar servicios en un entorno cloud. Recibe la especificación de la coreografía en un script que realiza todas las tareas necesarias para lanzar el servicio (procesos, preparación de servicios, etc.) [6].

Estas son las componentes que el servicio “Enactment Engine” tiene por defecto. Además incluye otras componentes de terceros como “Chef Solo”, encargada de instalar la configuración del middleware y sus componentes en un nodo concreto, y el “Cloud Gateway”, encargado de crear y destruir máquinas virtuales (nodos) [6] para el despliegue de los servicios.

## 7 Propiedades

Las principales propiedades de CHORéOS, como ya se ha dicho en secciones anteriores, son las siguientes:

- **Escalable:** La naturaleza distribuida de los servicios, hace que se puedan tener más nodos en momentos dados para hacer frente a la demanda de un servicio concreto. Además, el middleware tiene un mecanismo de escalabilidad que permite estimar el rendimiento de un servicio en función del tiempo de respuesta de los mismos.
- **Tolerante a fallos:** Al igual que antes, debido a la naturaleza distribuida, se pueden tener réplicas de servicios, los cuales pueden ser cambiados por el EE en caso de fallo.
- **Fiable y sencillo:** Ofrece un IDRE que permite crear coreografías de una manera más sencilla que si no se utilizase el middleware. Además se comprueba que tenga unos criterios de calidad mínimos antes de poder ser desplegado.
- **Interoperabilidad:** Al hacer uso de adaptadores, los distintos servicios heterogéneos pueden comunicarse sin problema, al traducir sus mensajes a un lenguaje común y luego ser vueltos a traducir en el otro extremo. [2]

## 8 Ejemplos de funcionamiento

La organización ha realizado tres ejemplos de uso de CHORéOS. “Passenger-Friendly Airport”, “Adaptive Customer Relationship Booster” y “DynaRoute”. [2] Yo me centraré solo en el primer ejemplo.

*Passenger-Friendly Airport* se concentra en los servicios suministrados a los pasajeros. Describe el uso de la información que aparece en un aeropuerto para adaptar los servicios ofrecidos a los viajeros, ofreciendo una mejor calidad. CHORéOS se usa para obtener acceso a los sensores, tanto del aeropuerto como en los dispositivos de los pasajeros. Por ejemplo, se usan micrófonos para ajustar automáticamente el volumen de las megafonías, dependiendo de que nivel de

sonido haya (si hay mucha gente, se sube el nivel de la megafonía automáticamente). [6]

Además, en los dispositivos de los pasajeros se instala un servicio de localización, para darles indicaciones de donde deben ir, haciendo que la navegación por el aeropuerto sea más sencilla. [6]

La prueba se hizo usando CHOReOS y una solución “ad-hoc”. Para la solución con CHOReOS, tardaron 40 minutos en realizar la especificación de la coreografía de servicios y 4 minutos en desplegarlo a nodos. Mientras que con la versión “ad-hoc” tardaron 9 horas para la especificación y una hora más para desplegarlo [5].

Esto demuestra que no solo es mejor usar CHOReOS, sino que además es más rápido de desarrollar usando este middleware.

## 9 Análisis crítico

Este middleware presenta una capa de abstracción muy interesante. El poder realizar servicios distribuidos a partir de otros servicios, también distribuidos. Además, incluye un entorno en el que definir las interacciones con otros servicios necesarias, haciendo que sea mucho más simple y general para definir.

Otro aspecto muy interesante es el hecho de tener una tolerancia a fallos en el propio middleware, algo muy importante cuando se trabaja en entornos distribuidos. También es agnóstico de la arquitectura y protocolos subyacente de los servicios, ofreciendo una interfaz general para que puedan interactuar entre ellos.

Además, parece ser que va a ser el dominante, ya que han aparecido más estándares, incluso antes que CHOReOS, pero que nunca han llegado a implementarse del todo, haciendo que este middleware sea el único en su campo.

La eficacia de este middleware está ya probada y documentada, haciendo que sea una muy buena opción para desarrollar una aplicación de servicios escalable y tolerante a fallos.

## 10 Conclusiones

Como conclusión, se puede decir que este middleware tiene mucho potencial para ser el dominante en su ámbito. Además, se puede ver que han tenido en cuenta cuestiones de heterogeneidad, detección de errores, escalabilidad, etc. Esto hace que tenga un rendimiento excelente, como aparece en algunos artículos.

Además, no solo proveen el propio middleware, también proveen un entorno de desarrollo (IDRE), haciendo que sea mucho más fácil el modelado y la verificación de los servicios.

Por último, como la tendencia es hacer uso cada vez más de los servicios, puede que sea una excelente herramienta para desarrollar aplicaciones basadas en servicios, o incluso desarrollar más servicios a partir de existentes.

## References

1. Hugues Vincent, Valérie Issarny, Nikolaos Georgantas, Emilio Francesquini, Alfredo Goldman, and Fabio Kon. 2010. CHOReOS: scaling choreographies for the internet of the future. In *Middleware '10 Posters and Demos Track (Middleware Posters '10)*. Association for Computing Machinery, New York, NY, USA, Article 8, 1–3. <https://doi.org/10.1145/1930028.1930036>
2. M. Autili, P. Inverardi and M. Tivoli, “CHOREOS: Large scale choreographies for the future internet,” 2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE), 2014, pp. 391-394, doi: 10.1109/CSMR-WCRE.2014.6747202.
3. M. A. Razzaque, M. Milojevic-Jevric, A. Palade and S. Clarke, “Middleware for Internet of Things: A Survey,” in *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70-95, Feb. 2016, doi: 10.1109/JIOT.2015.2498900.
4. Ben Hamida, Amira & Lockerbie, James & Bertolino, Antonia & Angelis, Guglielmo & Georgantas, Nikolaos & Pathak, Animesh & Bartkevicius, Rokas & Châtel, Pierre & Autili, Marco & Tivoli, Massimo & Di Ruscio, Davide & Zarras, Apostolos & Besson, Felipe & Santos, Carlos Eduardo & Cukier, Daniel & Ferreira Leite, Leonardo & Oliva, Gustavo & Ngoko, Yanik. (2011). Specification of the CHOReOS IDRE (D5.2).
5. L. Leite, C. E. Moreira, D. Cordeiro, M. A. Gerosa and F. Kon, "Deploying Large-Scale Service Compositions on the Cloud with the CHOReOS Enactment Engine," 2014 IEEE 13th International Symposium on Network Computing and Applications, 2014, pp. 121-128, doi: 10.1109/NCA.2014.25.
6. Amira Ben Hamida, Fabio Kon, Nelson Lago, Apostolos Zarras, Dionysis Athanassopoulos, et al.. Integrated CHOReOS middleware - Enabling large-scale, QoS-aware adaptive choreographies. 2013. hal-00912882

## 11 Autoevaluación

### 11.1 ¿La explicación es clara y el contenido está estructurado?

La explicación he intentado que sea lo más clara posible, intentando centrarme en los conceptos principales del middleware. Además, el contenido está estructurado, siguiendo las secciones que las normas del trabajo indican.

### 11.2 ¿Queda reflejado que se han estudiado y entendido las cuestiones de diseño, mecanismos, protocolos y servicios para la comunicación y coordinación?

Creo que sí, pero debido al número de páginas máximo (que al final he tenido que usar una página más), creo que no quedan reflejados absolutamente todos los aspectos estudiados, pero los conceptos principales sí se han entendido.

### 11.3 ¿Se incluyen ejemplos que ayudan a explicar la descripción de la tecnología middleware seleccionada?

Solo se incluye un ejemplo real en la sección destinada a ello. En ella se puede ver como se usa el middleware para resolver el problema, además de poder ver

la velocidad de desarrollo que se consigue con este. En las demás secciones he intentado ser lo más descriptivo posible, pero sin llegar a poner ejemplos.

**11.4 ¿Queda reflejado en el trabajo que se ha buscado y analizado suficiente bibliografía adicional y se han incluido las referencias encontradas?**

Sí. He investigado todo lo que he podido, incluyendo artículos más extensos donde se explica el proyecto en más profundidad. Además, todo lo que he usado ha sido referenciado en la bibliografía.

**11.5 ¿Se incluye un análisis crítico positivo y/o negativo de la tecnología middleware escogida y unas conclusiones como resultado del estudio y explicación realizadas?**

Sí. El análisis crítico ha sido positivo, ya que me parece una tecnología muy buena para poder reusar servicios existentes para crear otros. Además, he incluido también una conclusión sobre todo lo escrito y estudiado.

**11.6 ¿El informe realizado me ha llevado tiempo y servido para comprender la solución al problema abordado?**

Sí. Me ha requerido mucho tiempo realizar el informe, ya que he buscado muchos artículos para intentar comprender como funcionaba, ya que al principio no conseguía ver el problema que describían. Además, me ha servido para entender mejor la solución al problema, el cual se va a ir agravando cada vez más.

**11.7 ¿Se ha realizado alguna aportación adicional más allá de lo que en esencia pedía este trabajo?**

No. Principalmente por la restricción de páginas, ya que había mucho que contar y explicar, por lo que me he ceñido a lo que se pedía.

Por todo esto, pienso que mi calificación es de **6,0**.