Gómez Muñoz, L., Abrahao, S., Insfran, E.: Hacia un framework de adaptación para micro-servicios y contenedores. In: Fernández Montes, P. (ed.) Actas de las XVI Jornadas de Ciencia e Ingeniería de Servicios (JCIS 2021). Sistedes (2021). https://hdl.handle.net/11705/JCIS/2021/038

Hacia un Framework de Adaptación para Micro-servicios y Contenedores*

Luisa Gómez-Muñoz, Silvia Abrahão, y Emilio Insfran

Instituto Universitario Mixto de Tecnología Informática
Universitat Politècnica de València, Valencia, España
luigomuo@upv.edu.es, sabrahao@dsic.upv.es, einsfran@dsic.upv.es

Resumen Las empresas de software requieren soluciones y procesos flexibles que sirvan de apoyo en el desarrollo y despliegue de servicios y faciliten la adaptación a lo largo de su ciclo de vida. En este escenario surgen las arquitecturas orientadas a micro-servicios, donde las aplicaciones en la nube se dividen en varios servicios ligeros, usualmente empaquetados en contenedores, para su implementación. Con el propósito de realizar una contribución en este campo, se presenta una propuesta inicial de un framework para la adaptación de sistemas software basados en micro-servicios y contenedores soportado en un proceso DevOps. Este framework provee una guía para seleccionar tareas, técnicas y herramientas a ser utilizadas durante el diseño y mantenimiento del sistema con el fin de dotarlo de capacidades adaptativas específicas. El framework está basado en la arquitectura MAPE-K y fue definido a partir de los resultados de una revisión sistemática sobre mecanismos de adaptación en arquitecturas de micro-servicios.

Palabras claves: Adaptación · micro-servicios · contenedores · DevOps

1. Introducción

El paradigma de micro-servicios es uno de los patrones arquitectónicos más utilizados en los últimos años para garantizar la escalabilidad, flexibilidad y evolución de las aplicaciones de software. Esta arquitectura permite el desacoplamiento de las distintas partes de un sistema altamente complejo mediante la definición de componentes independientes que implementan funcionalidades específicas [12] y tienen la capacidad de evolucionar como servicios individuales [7]. El desacoplamiento de componentes en la arquitectura de micro-servicios conduce naturalmente a la contenerización, dada la flexibilidad que brinda en la fase de implementación.

El potencial que brindan estas tecnologías es ahora de particular interés para la creación de sistemas de software adaptativos. Sin embargo, el esfuerzo en este sentido se ha centrado principalmente en resolver problemas de asignación

* Este trabajo ha sido financiado por los proyectos Adapt@Cloud (TIN2017-84550-R) y UX-Adapt (AICO/2020/113)

Gómez Muñoz, L., Abrahao, S., Insfran, E.: Hacia un framework de adaptación para micro-servicios y contenedores. In: Fernández Montes, P. (ed.) Actas de las XVI Jornadas de Ciencia e Ingeniería de Servicios (JCIS 2021). Sistedes (2021). https://hdl.handle.net/11705/JCIS/2021/038

dinámica de recursos en las plataformas de computación en la nube. Además, no existe una línea clara sobre las tecnologías y estrategias existentes para incorporar cualidades adaptativas a los sistemas de software orientados a micro-servicios y desplegados en contenedores. En este trabajo, se presenta un planteamiento inicial de un framework para la adaptación de micro-servicios desplegados en contenedores y soportado en un proceso DevOps [4]. El framework está basado en la arquitectura MAPE-K y fue definido a partir de los resultados de una revisión sistemática sobre mecanismos de adaptación en arquitecturas de micro-servicios.

El resto de este documento está organizado de la siguiente manera: La Sección 2 discute los trabajos relacionados. En la Sección 3 se presenta el método seguido para realizar este trabajo. En la Sección 4 se presenta el marco para la adaptación propuesto y los resultados de una evaluación preliminar de su uso. En la Sección 5 se presentan las conclusiones obtenidas y las líneas de trabajo futuras.

2. Trabajos Relacionados

La capacidad de adaptación de un sistema se traduce en la ejecución de una serie de procesos para extender o modificar sus dimensiones con el objetivo de habilitar o mejorar funcionalidades, interacciones y aspectos de comunicación entre los componentes del sistema y su entorno [6]. Durante los últimos años, los estudios en contextos *cloud* o micro-servicios se han enfocado en la adaptación orientada a características de calidad o requerimientos no funcionales de los sistemas. Fokaefs et al. en [3] presentan los resultados de un estudio empírico para la evaluación de tres métodos de adaptación sobre una aplicación Web y su impacto sobre la efectividad en el manejo de recursos de CPU en las máquinas que alojan la aplicación. Pahl et al. en [10] realizan una clasificación taxonómica de los estudios orientados a la implementación de tecnologías basadas en contenedores en entornos *cloud* enfocando el análisis del componente arquitectónico en mecanismos de adaptación en términos de recursos y encontrando un balance en conjunto con los objetivos de nivel de servicio y los parámetros de calidad de la infraestructura.

Entre los estudios que orientan su investigación a la evaluación de la adaptación en términos funcionales, Jamshidi et al. [5] presentan una clasificación taxonómica de las investigaciones con enfoque en la evolución del software centrado en la arquitectura. Macías-Escrivá et al. [8] abordan los principales conceptos, herramientas, métodos y aplicaciones que habilitan la capacidad de auto-adaptabilidad desde la perspectiva computacional. Sin embargo, en la mayoría de los trabajos no se considera en la búsqueda los conceptos de micro-servicios o contenedores. La investigación de Bogner et al. [2] reúne las diferentes tácticas de modificabilidad arquitectónica para la evolución de sistemas en tres categorías: aumento de la cohesión, reducción del acoplamiento y aplazamiento del tiempo de enlace. Esta lista es comparada con ocho principios de los micro-servicios con el fin de identificar en qué medida se encuentran los dos conceptos relacionados

Gómez Muñoz, L., Abrahao, S., Insfran, E.: Hacia un framework de adaptación para micro-servicios y contenedores. In: Fernández Montes, P. (ed.) Actas de las XVI Jornadas de Ciencia e Ingeniería de Servicios (JCIS 2021). Sistedes (2021). https://hdl.handle.net/11705/JCIS/2021/038

y los patrones de diseño que se pueden utilizar, encontrando que la mayoría de las relaciones están orientadas a la reducción del acoplamiento.

3. Método

Para la definición del framework de adaptación se ha realizado una revisión sistemática de la literatura (RSL) con el objetivo de recopilar y clasificar taxonómicamente las aproximaciones propuestas para adaptar micro-servicios desplegados en contenedores. La revisión se realizó en tres fases: *planificación, ejecución* y *documentación*, cuyos principales aspectos se resumen a continuación:

Preguntas de Investigación: El objetivo de la RSL consiste en *identificar las estrategias de adaptación aplicadas en la construcción de sistemas software cuya arquitectura esté basada en micro-servicios o hagan uso de contenedores*. Dicho objetivo orienta la formulación de las siguientes preguntas: (PI1) ¿Qué tipos de propuestas fueron planteadas para soportar la adaptación de sistemas software basados en micro-servicios o contenedores?; (PI2) ¿Qué tipos de adaptación se presentan en sistemas software basados en micro-servicios o contenedores?; (PI3) ¿Qué factores motivan la necesidad de adaptación en sistemas software basados en micro-servicios o contenedores?; (PI4) ¿Qué técnicas y herramientas se han empleado en la adaptación de sistemas software utilizando micro-servicios o contenedores?; (PI5) ¿Qué tipo de evaluación se ha realizado para validar las aproximaciones, técnicas y/o herramientas propuestas?

Protocolo de Búsqueda: La búsqueda se realizó durante el mes de enero de 2020 en cuatro motores de búsqueda: *IEEE Xplore*, *ACM Digital Library*, *Science Direct* y *Springer Link*, utilizando la siguiente cadena de búsqueda: *((adapt*) and ((microservice* or micro-service* or "micro service") or (container* or kubernet* or docker)))*. Para la construcción de la cadena de búsqueda se tuvieron en cuenta palabras clave relacionadas a los principales conceptos: (i) adaptación, adaptabilidad o auto-adaptación, en inglés *adaptive* o *adaptation* o *auto-adaptation*, se utilizó el término *adapt** para cubrirlos a todos. (ii) microservicios, en este caso se identificaron tres posibles formas de mencionarlo en los artículos *microservice*, *micro-service* o *micro service*, el operador *OR* nos permite incluir las tres opciones. (iii) contenedores, en este caso incluimos *container** o las herramientas más utilizadas *Docker* o *Kubernetes*.

Estrategia de selección: En la selección de estudios primarios se tuvo en cuenta artículos científicos que presentan aproximaciones o herramientas utilizadas para la adopción o implementación de mecanismos de adaptación en sistemas basados en micro-servicios o contenedores en el periodo comprendido entre 2012 y enero de 2020. También estudios que comparan o evalúan las aproximaciones o herramientas propuestas dentro del mismo marco. Tras la ejecución de este proceso, fueron seleccionados 13 estudios primarios. El listado completo se encuentra disponible en este enlace *https://tinyurl.com/jcis2021-estudios*.

Evaluación de calidad: Se realizó la evaluación de los artículos aplicando criterios orientados a la calidad del estudio para identificar la pertinencia del artículo respecto al objetivo de la revisión sistemática planteada. Como resul-

Gómez Muñoz, L., Abrahao, S., Insfran, E.: Hacia un framework de adaptación para micro-servicios y contenedores. In: Fernández Montes, P. (ed.) Actas de las XVI Jornadas de Ciencia e Ingeniería de Servicios (JCIS 2021). Sistedes (2021). https://hdl.handle.net/11705/JCIS/2021/038

tado de este proceso fueron descartados dos de los artículos primarios, debido principalmente a que no presentaban ninguna evaluación del enfoque propuesto.

Extracción de datos: Los criterios de extracción de los datos e información de los estudios fueron analizados en conformidad con las preguntas de investigación planteadas y agrupados en los siguientes temas:

1 **Aproximaciones (PI1)**: Clasifica los tipos de aproximación propuestos.
2 **Tipo de adaptación (PI2)**: Consolida los criterios que caracterizan el tipo de adaptación implementado o propuesto en el estudio.
3 **Motivadores de adaptación (PI3)**: Identifica las necesidades en el sistema que impulsaron la adopción de mecanismos de adaptación.
4 **Técnicas de adaptación (PI4)**: Recopila los aspectos técnicos utilizados en los estudios para conseguir la adaptación del sistema.
5 **Herramientas (PI4)**: Recoge las herramientas tecnológicas utilizadas en cada uno de los estudios para soportar la adaptación.
6 **Modelo de proceso de desarrollo (PI4)**: Recopila los modelos de proceso de desarrollo adoptados en los estudios.
7 **Evaluación del nivel de madurez (PI5)**: Agrupa los métodos de evaluación empleados para proporcionar evidencias sobre la utilidad de las aproximaciones y herramientas propuestas.

3.1. Clasificación Taxonómica

Con la finalidad de proporcionar un marco de referencia que permita identificar de manera ágil y sencilla los aspectos básicos de la adaptabilidad en sistemas de software basados en micro-servicios se planteó una estructura taxonómica, resultante de la extracción y clasificación de datos relacionados con el proceso de adaptación empleado en los estudios primarios. La Tabla 1 presenta dicha taxonomía, estructurada en tres niveles: (i) *clase*, corresponde a la agrupación principal; (ii) *criterio de adaptación*, determinado en conformidad con la exploración realizada sobre los principales aspectos de adaptación; así mismo, algunos criterios pueden tener (iii) *subcriterios de adaptación* con los que se pretende agregar mayor detalle para la comprensión de la taxonomía. Por último se presentan las opciones identificadas en los estudios primarios para cada uno de los atributos de adaptación. Los criterios se señalan con el código ECn.m, en el que **EC** corresponde a las siglas en inglés para criterio de extracción (*Extraction Criteria*), **n** es el identificador numérico de la clase que agrupa los criterios y **m** el identificador numérico del criterio.

La taxonomía planteada permite posicionar los trabajos existentes y futuros en cuanto a los distintos atributos y mecanismos de adaptabilidad que poseen, así como su expresividad.

4. Framework de Adaptación Propuesto

En esta sección, se propone un marco tecnológico para la adaptación de micro-servicios existentes, así como los artefactos y reglas que lo soportan con base en los resultados obtenidos en la revisión sistemática de la literatura.

Gómez Muñoz, L., Abrahao, S., Insfran, E.: Hacia un framework de adaptación para micro-servicios y contenedores. In: Fernández Montes, P. (ed.) Actas de las XVI Jornadas de Ciencia e Ingeniería de Servicios (JCIS 2021). Sistedes (2021). https://hdl.handle.net/11705/JCIS/2021/038

Tabla 1: Clasificación taxonómica

CLASE	CRITERIO		OPCIONES
EC1 Aproximaciones	EC1.1 Tipo de propuesta		Método, framework, Herramienta, Modelo, Arquitectura, Enfoque /Aproximación
	EC1.2 Dominio de aplicación		Cloud Computing, Sistemas Distribuidos, SOA, IoT, Ciudades Inteligentes, i-CSS (Intelligent Complex Software Systems)
	EC1.3 Contexto de uso		Académico, Industrial
EC2 Tipo de Adaptación	EC2.1 Fase de adaptación		Diseño, Integración, Despliegue, Operación, Monitorización
	EC2.2 Nivel de autonomía		Completamente autónomo, Parcialmente autónomo, Humano en medio
	EC2.3 Tiempo de adaptación		Diseño (Estático), Carga, Ejecución (Dinámico)
	EC2.4 Mecanismo de adaptación		Abierto, Cerrado
	EC2.5 Actividad de adaptación		Reactiva, Proactiva
	EC2.6 Condiciones de adaptación		Auto-configuración, Auto-optimización, Auto-curación, Auto-protección
EC3 Motivadores	EC3.1 Propósito de adaptación		Correctivo, Perfectivo, Adaptativo, Preventivo
	EC3.2 Necesidades de adaptación		QoS (Calidad del Servicio), FT (Tolerancia a fallos), FD (Detección de fallos), NR (Nuevos Requerimientos), Optimización de recursos, SLA (Acuerdos de nivel de servicio).
EC4 Técnicas de adaptación	EC4.1 Frecuencia de adaptación		Continúa, Periódica, Arbitraria
	EC4.2 Lógica de adaptación		Probabilística, Basada en reglas, Basada en casos, Basada en lógica, Basada en ontologías, Basada en evidencia, Basada en lógica difusa
EC5 Herramientas	EC5.1 Herramientas de Soporte	EC5.1.1 Monitorización	EnviroSCALE, ConMon, Zipkin, Zuul, Logstash, Heapster, MetroFunnel
		EC5.1.2 Despliegue	Docker, Kubernetes, Fleet, Dynamite
		EC5.1.3 Pruebas	Travis
		EC5.1.4 Almacenamiento	Etcd, Influxdb, Elasticsearch, Eureka
		EC5.1.5 Integración	Eureka, Open vSwitch, Axis2
		EC5.1.6 Gestión	Activiti
	EC5.2 Protocolo / Estilo Arquitectura		REST, MQTT, SOAP
EC6 Modelo de proceso	EC6.1 Modelo de proceso		DevOps
EC7 Evaluación	EC7.1 Método de evaluación		Experimento, Encuesta, Caso de estudio

De acuerdo con la taxonomía presentada, las principales necesidades de adaptación corresponden con *mejora de la calidad del servicio, tolerancia a fallos, incorporación de nuevos requerimientos al sistema y optimización de recursos.* Este listado de motivadores proporciona una hoja de ruta que define los propósitos de adaptación a abordar. A partir de los mismos se clasificaron las tareas, técnicas y herramientas identificadas en los estudios primarios seleccionados y las relaciones entre ellas en concordancia con las fases del bucle MAPE-K: Monitorización(M), Análisis(A), Planificación(P) y Ejecución(E), sobre una base de conocimiento K, considerado un modelo de referencia para sistemas auto adaptativos [1]. El resultado obtenido se puede observar en la Figura 1 y es descrito a continuación.

Necesidad de adaptación: El framework propuesto parte de la necesidad de adaptación que se desea cubrir en el sistema, que se pueden clasificar entre:

- **Mitigación de fallos**: Reúne las necesidades relacionadas con la prevención, detección y corrección de fallos [11]
- **Requerimientos no funcionales - QoS**: Reúne las necesidades relacionadas con el cumplimiento, incorporación o mejora de requerimientos no funcionales, mayormente, aquellos orientados a la calidad del servicio.
- **Nuevos requerimientos**: La adaptación del micro-servicio se realiza con el objetivo de incorporar nuevos requerimientos de negocio o mejorar los existentes de forma evolutiva.
- **Optimización de recursos**: La adaptación del micro-servicio se realiza con el objetivo de mejorar o mantener el desempeño de los recursos de hardware y software que lo componen.

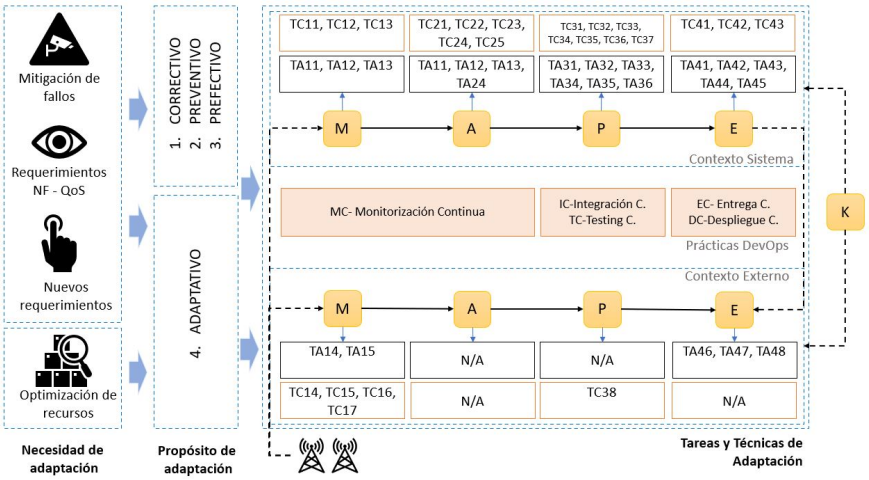

Figura 1: framework de adaptación

Contexto: Si bien es posible que una adaptación impacte varios contextos independientemente del propósito, en este framework se establece la guía hacia dos contextos en particular: (a) **Contexto externo**: Necesidades de adaptación relacionadas con la optimización de recursos sobre los componentes de la capa de infraestructura; contenedores, máquinas virtuales, sensores y demás elementos de hardware o software que no son parte de la propia aplicación. (b) **Contexto sistema**: Elementos activados durante la ejecución de una o varias instancias de la aplicación o servicio. En este caso, el propósito de adaptación es indiferente, pero las necesidades se limitan a aquellas que no conllevan la optimización de recursos físicos o virtuales externos al sistema.

Propósito de adaptación: El framework propone la definición del propósito de adaptación en correspondencia con la necesidad previamente identificada.

Tareas para la adaptación: Las tareas de adaptación corresponden al conjunto de pasos necesarios para llevar a cabo el objetivo de la fase del flujo al cual pertenecen. En la Tabla 2 se listan las tareas que fueron identificadas en cada uno de los estudios primarios en relación con el contexto y fase del flujo impactado. Cada una se encuentra señalada con las siglas TA (Tarea de adaptación), el número de fase (1: Monitorización, 2: Análisis, 3: Planificación: 4: Ejecución) y un número consecutivo. Pueden ser seleccionadas en función del propósito, condición, actividad y/o mecanismo de adaptación que se desee emplear. Normalmente las tareas relacionadas a las fases de monitorización y análisis están determinadas por el tipo de actividad, mientras que las tareas de planificación y ejecución se establecen con base en las condiciones y mecanismos de adaptación.

Técnicas para la adaptación: Las técnicas de adaptación comprenden los mecanismos o estrategias empleados por los diferentes estudios primarios para lograr la adaptación de micro-servicios. En la Tabla 3 se muestra el listado com-

Gómez Muñoz, L., Abrahao, S., Insfran, E.: Hacia un framework de adaptación para micro-servicios y contenedores. In: Fernández Montes, P. (ed.) Actas de las XVI Jornadas de Ciencia e Ingeniería de Servicios (JCIS 2021). Sistedes (2021). https://hdl.handle.net/11705/JCIS/2021/038

Gómez Muñoz, L., Abrahao, S., Insfran, E.: Hacia un framework de adaptación para micro-servicios y contenedores. In: Fernández Montes, P. (ed.) Actas de las XVI Jornadas de Ciencia e Ingeniería de Servicios (JCIS 2021). Sistedes (2021). https://hdl.handle.net/11705/JCIS/2021/038

Tabla 2: Tareas para la adaptación

FASE	CONTEXTO	TAREA
Monitorización	Sistema	TA11-Detección de eventos
		TA12-Captura de propiedades
		TA13-Captura de muestras
	Externo	TA14-Captura de mensajes
		TA15-Carga de eventos
Análisis	Sistema	TA21-Evaluación de errores
		TA22-Predicción de fallos
		TA23-Cálculo de carga de trabajo
		TA24-Análisis de probabilidad
Planificación	Sistema	TA31-Evaluación de errores
		TA32-Replanificación de ruta
		TA33-Planificación de reconfiguración
		TA34-Identificación de acciones
		TA35-Generación de casos de prueba
		TA36-Validación previa del plan de adaptación
Ejecución	Sistema	TA41-Creación de proceso de negocio
		TA42-Ejecución de acción
		TA43-Ejecución de casos de prueba
		TA44-Reconfiguración de componentes
		TA45-Inicialización de un componente
	Externo	TA46-Despliegue de contenedor
		TA47-Orquestación de contenedores
		TA48-Eliminación o detención de un contenedor

pleto de técnicas según la fase del flujo MAPE-K a la que apoya, el contexto en el que se aplica y la lógica de adaptación empleada por la técnica para control de la adaptación; (i)Lógica: adaptación basada en lógica de predicados, (ii)Reglas: basada en un conjunto de reglas o políticas, (iii)Probabilística: basada en un modelo probabilístico. Cada técnica se encuentra señalada con las siglas TC, el número de fase (1: Monitorización, 2: Análisis, 3: Planificación: 4: Ejecución) y el número consecutivo correspondiente.

Herramientas para la adaptación: En la Tabla 1, en el criterio EC5.1 se listan las herramientas según las fases del proceso DevOps que apoyan. Son incorporadas al framework en correspondencia con una fase del flujo MAPE-K. La etapa de *monitorización* emplea las herramientas del subcriterio EC5.1.1; el *análisis* contempla EC5.1.3; en *planificación* se utilizan las herramientas del subcriterio EC5.1.6 y en *ejecución*, aquellas de EC5.1.2 y EC5.1.5. Las herramientas del subcriterio EC5.1.4 son empleadas como *bases de conocimiento*.

Base de Conocimiento: La base de conocimiento del flujo MAPE-K se encarga de almacenar datos relevantes respecto al comportamiento del sistema, parámetros de configuración, resultados de ejecución de determinadas actividades y en algunos casos decisiones relacionadas con la adaptación.

Finalmente, para hacer uso de este framework se propone una guía que establece las actividades que deberían llevarse a cabo para incorporar capacidades de adaptación en el diseño de un sistema basado en micro-servicios y desplegado en contenedores. Estas actividades son las siguientes: **(1) Definir la necesidad de adaptación**, de acuerdo con la taxonomía propuesta en el framework de clasificación; **(2) Definir el propósito de adaptación**, según la necesidad de adaptación identificada; **(3) Identificar la fase de adaptación**, se debe esta-

Gómez Muñoz, L., Abrahao, S., Insfran, E.: Hacia un framework de adaptación para micro-servicios y contenedores. In: Fernández Montes, P. (ed.) Actas de las XVI Jornadas de Ciencia e Ingeniería de Servicios (JCIS 2021). Sistedes (2021). https://hdl.handle.net/11705/JCIS/2021/038

Tabla 3: Técnicas para la adaptación

FASE	CONTEXTO	TÉCNICA	LÓGICA
Monitorización	Sistema	TC11-*Interaction Daemon Container*	Reglas
		TC12-Registro de eventos históricos	Reglas
		TC13-Funciones de monitorización adyacentes	Reglas
	Externo	TC14-Reducir volumen de muestra	Lógica
		TC15-Reducir frecuencia de muestreo	Lógica
		TC16-Inyección datos prueba	Reglas
		TC17-Recopilar información desde orquestadores	Reglas
Análisis	Sistema	TC21-Procesamiento eventos complejos	Probabilística
		TC22-Aprendizaje en línea	Probabilística
		TC23-Obtener vista parcial del sistema	Reglas
		TC24-Comparación con gráficos	Reglas
		TC25-Cálculo afinidad entre micro-servicios	Reglas
Planificación	Sistema	TC31-Planificación local	Lógica
		TC32-Planificación global	Lógica
		TC33-Selección políticas según su peso	Probabilística
		TC34-Evaluación adaptativa de confiabilidad	Probabilística
		TC35-Planeación de afinidad heurística	Reglas
		TC36-Planeación de afinidad óptima	Reglas
		TC37-Modelos en tiempo de ejecución	Reglas
	Externo	TC38-Decisiones con base en flujos de actividades	Reglas
Ejecución	Sistema	TC41-Recuperación hacia atrás	Probabilística
		TC42-Recuperación hacia adelante	Probabilística
		TC43-Determinar puntos de sincronización	Reglas

blecer la fase del flujo de retroalimentación MAPE-K (Monitorización, Análisis, Planificación, Ejecución) que debería incorporar la estrategia de adaptación; **(4) Seleccionar actividad DevOps**, una vez identificada la fase de adaptación, se podrá determinar la o las actividades continuas del proceso DevOps (Monitorización continua, Integración continua, etc.) en las que se incorporarán las capacidades de adaptación; **(5) Seleccionar tareas y técnicas por fase**, consiste en elegir las tareas y técnicas que mejor se acoplen en cada una de las fases del flujo MAPE, y **(6) Seleccionar herramientas**, se deben seleccionar las herramientas considerando las tareas, técnicas y actividad DevOps a soportar.

4.1. Prueba de Concepto

Con el objetivo de ilustrar la aplicación del framework de adaptación, se usó como escenario de prueba el *M3 System* [9], cuyo diseño es una propuesta para la monitorización de micro-servicios desplegados en contenedores. El sistema consta de dos componentes principales: el *administrador de monitorización* y el *agente de monitorización*. Para extender su funcionalidad, se plantea un objetivo de adaptación dinámica: *La capacidad de identificar posibles fallos sobre los agentes de monitorización y recuperar su operatividad en caso de interrupción del servicio.*

Se identifica como necesidad de adaptación la **mitigación de fallos** en tiempo de ejecución. El propósito de adaptación en este caso, puede ser tanto **preventivo** como **correctivo** ya que se pretende detectar posibles incidentes pero también apoyar la recuperación del sistema ante un eventual fallo. Según el objetivo establecido, la adaptación debe ser dinámica, por lo que la fase en la cual el sistema realizará la adaptación será en **operación** y por lo tanto, se sugiere la incorporación de actividades de **entrega, testing y monitorización continuos**. La necesidad y propósito de adaptación identificados nos guían hacia la

integración de tareas, técnicas y herramientas en el **contexto de sistema**. En la Figura 2 se muestra el sistema resultante con las modificaciones sobre el *M3 System* que incorpora diferentes tareas, técnicas y herramientas sobre la base de un flujo MAPE-K.

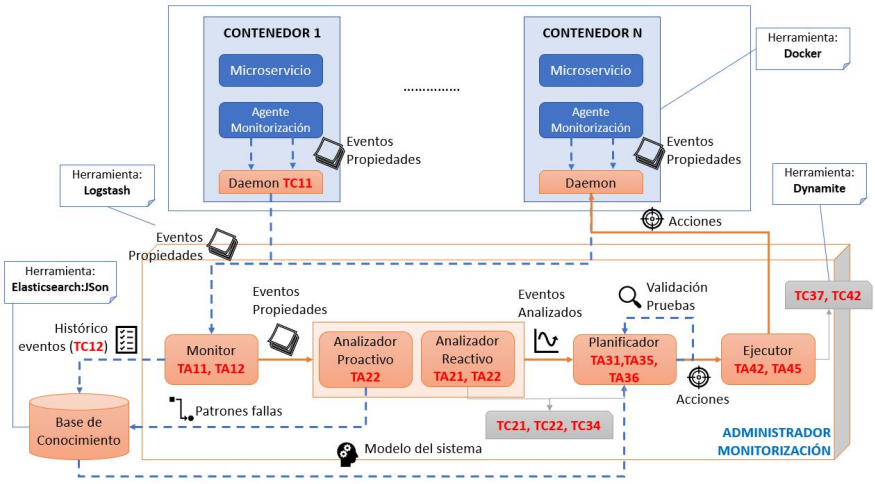

Figura 2: Diagrama del *M3 System* con capacidades de adaptación

5. Conclusiones y Trabajo Futuro

En este trabajo se ha presentado una versión preliminar de un framework para guiar el proceso de adaptación de sistemas software basados en micro-servicios y desplegados en contenedores para procesos DevOps. El framework se ha definido en base a los resultados de una revisión sistemática que ha recopilado las tareas, técnicas y herramientas de adaptación. Este framework puede servir como hoja de ruta para arquitectos y diseñadores de este tipo de sistemas ya que incorpora una estrategia para la selección de las actividades del proceso DevOps aplicables en conformidad con la fase y el propósito de adaptación del sistema.

El resultado obtenido nos ha aportado evidencias sobre la viabilidad del uso del framework en un sistema con poca complejidad. Sin embargo, los conceptos presentados en este framework deben operacionalizarse y estudiarse para otros casos de adaptación específicos. Una limitación del framework propuesto es que se basa en un número reducido de estudios. Actualmente estamos actualizando la revisión sistemática con otros trabajos publicados recientemente, lo que nos permitirá validar y proporcionar nuevas evidencias sobre la adecuación y utilidad de las tareas, técnicas y herramientas planteadas. Un trabajo futuro consiste en la validación del framework mediante la realización de otros casos de estudio más realistas.

Gómez Muñoz, L., Abrahao, S., Insfran, E.: Hacia un framework de adaptación para micro-servicios y contenedores. In: Fernández Montes, P. (ed.) Actas de las XVI Jornadas de Ciencia e Ingeniería de Servicios (JCIS 2021). Sistedes (2021). https://hdl.handle.net/11705/JCIS/2021/038

Gómez Muñoz, L., Abrahao, S., Insfran, E.: Hacia un framework de adaptación para micro-servicios y contenedores. In: Fernández Montes, P. (ed.) Actas de las XVI Jornadas de Ciencia e Ingeniería de Servicios (JCIS 2021). Sistedes (2021). https://hdl.handle.net/11705/JCIS/2021/038

Referencias

1. Arcaini, P., Riccobene, E., Scandurra, P.: Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation. In: Proceedings - 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2015. pp. 13–23. Institute of Electrical and Electronics Engineers Inc. (8 2015)
2. Bogner, J., Wagner, S., Zimmermann, A.: Using architectural modifiability tactics to examine evolution qualities of Service- and Microservice-Based Systems: An approach based on principles and patterns. In: Software-Intensive Cyber-Physical Systems. vol. 34, pp. 141–149. Springer (2019)
3. Fokaefs, M., Rouf, Y., Barna, C., Litoiu, M.: Evaluating Adaptation Methods for Cloud Applications: An Empirical Study. In: IEEE International Conference on Cloud Computing, CLOUD. vol. 2017-June, pp. 632–639. IEEE (2017)
4. Gómez, L.F.: Tesis de Máster, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. Valencia (Septiembre 2020), `https://riunet.upv.es/handle/10251/151747`
5. Jamshidi, P., Ghafari, M., Ahmad, A., Pahl, C.: A framework for classifying and comparing architecture-centric software evolution research. In: Proceedings of the European Conference on Software Maintenance and Reengineering, CSMR (2013)
6. Kell, S.: A survey of practical software adaptation techniques. Journal of Universal Computer Science **14**(13), 2110–2157 (2008)
7. Liu, P., Mao, X., Zhang, S., Hou, F.: Towards Reference Architecture for a Multi-layer Controlled Self-adaptive Microservice System NSFC project View project Self-adaptive microservice syetem View project Towards Reference Architecture for a Multi-layer Controlled Self-adaptive Microservice System (2018)
8. Macías-Escrivá, F.D., Haber, R., del Toro, R., Hernandez, V.: Self-adaptive systems: A survey of current approaches, research challenges and applications. Expert Systems with Applications **40**(18), 7267–7279 (2013)
9. Noor, A., Jha, D.N., Mitra, K., Jayaraman, P.P., Souza, A., Ranjan, R., Dustdar, S.: A framework for monitoring microservice-oriented cloud applications in heterogeneous virtualization environments. In: IEEE Int. Conference on Cloud Computing, CLOUD. vol. 2019-July, pp. 156–163. IEEE Computer Society (2019)
10. Pahl, C., Brogi, A., Soldani, J., Jamshidi, P.: Cloud Container Technologies: a State-of-the-Art Review. IEEE Transactions on Cloud Computing (2017)
11. Power, A., Kotonya, G.: A Microservices Architecture for Reactive and Proactive Fault Tolerance in IoT Systems. In: 19th IEEE Int. Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2018. IEEE (2018)
12. Zhang, S., Zhang, M., Ni, L., Liu, P.: A multi-level self-adaptation approach for microservice systems. In: 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analytics, ICCCBDA 2019. pp. 498–502. IEEE (2019)