

Exploración de Técnicas de Machine Learning para Migración de Sistemas Legados hacia Microservicios

Guillermo Rodríguez¹ Leonardo Da Rocha Araujo¹ Fabio Rocha²
Rodrigo Pereira dos Santos³

¹ ISISTAN (UNICEN-CONICET), Tandil, Argentina.

² UNIT, Aracaju, Sergipe, Brasil.

³ UNIRIO, Rio de Janeiro, Brasil.

{guillermo.rodriguez, leonardo.araujo}@isistan.unicen.edu.ar, fabio.gomes@souunit.com.br,
rps@uniriotec.br

RESUMEN

Actualmente, cada vez más empresas están adoptando microservicios para modernizar sus productos y tomar ventaja de sus prometidos beneficios como: agilidad, escalabilidad e integración continua, entre otros. Por un lado, los sistemas basados en microservicios presentan una arquitectura flexible y con alta capacidad de evolución. Sin embargo, por otro lado, hay desafíos técnicos (por ej. automatización de la infraestructura y *debugging* distribuido) y organizacionales (por ej. creación de equipos de trabajos *cross-functional*) que necesitan ser abordados.

Lamentablemente, migrar una arquitectura orientada a microservicios no es una tarea simple. En este proceso, los servicios pueden escalar más eficientemente y los ciclos de entregas se acortan debido al continuo despliegue. Normalmente, estas decisiones de diseño quedan sujetas a la intuición de desarrolladores y/o arquitectos, pero carecen de un análisis sistemático que les facilite la evaluación de alternativas y toma de decisiones. En este contexto, las técnicas de *machine learning* podrían contribuir a facilitar la exploración de diferentes alternativas de descomposición de arquitecturas de software en microservicios.

Palabras clave: *Arquitecturas de Microservicios, Sistemas Legados, Migración, Aprendizaje Automático, SOA.*

CONTEXTO

La propuesta emerge del proyecto PICT Joven (PICT-2018-01456, Migración de arquitecturas monolíticas hacia microservicios y contenedores) otorgado por la Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT), a través del Fondo para la Investigación Científica y Tecnológica (FONCyT). El mismo se desarrolla dentro del Instituto Superior de Ingeniería de Software Tandil (ISISTAN-CONICET) de la Facultad de Ciencias Exactas (EXA), Universidad Nacional del Centro de la provincia de Buenos Aires (UNCPBA), y unidad ejecutora del CONICET.

Colaboran con el proyecto miembros de Universidades extranjeras:

- Universidade Federal do Estado do Rio de Janeiro (UNIRIO), Rio de Janeiro, Brasil.
- Universidade Tirandentes (UNIT), Sergipe, Brasil.

1. INTRODUCCIÓN

En el entorno dinámico del mundo de hoy, las aplicaciones de software necesitan responder rápidamente a las demandas del entorno y ser lo más ágiles posible. Las aplicaciones necesitan estar alineadas con necesidades de negocio cambiantes, a las cuales se le suma una diversidad de clientes, como navegadores de escritorio y móviles, como así también aplicaciones móviles nativas y APIs de terceras partes. Dado que es difícil cumplir estos requisitos mediante el uso de aplicaciones monolíticas, el estilo arquitectónico cliente-servidor monolítico está siendo reemplazado por el estilo de microservicios (Rahman, 2018). Cada vez más empresas están adoptando este estilo

arquitectónico para modernizar sus productos y tomar ventaja de beneficios tales como: agilidad, escalabilidad, e integración continua (Di Francesco, 2018). Estas ventajas se perciben como superadoras respecto a arquitecturas orientadas a servicios convencionales.

La definición más aceptada del concepto de microservicios apunta a un enfoque para desarrollar una aplicación vista como un conjunto de pequeños servicios, cada uno corriendo en su propio proceso y comunicándose con recursos ligeros como HTTP (Fowler, 2018). El estilo de Microservicios no sólo trae aparejados beneficios, sino también desafíos. Por un lado, los sistemas basados en microservicios presentan una arquitectura flexible y con alta capacidad de evolución. Sin embargo, hay varios desafíos técnicos (por ej. automatización de la infraestructura y *debugging* distribuido) y organizacionales (por ej. creación de equipos de trabajos *cross-functional*) que necesitan ser abordados. Por ejemplo, las cuestiones técnicas pueden ser que el sistema está altamente acoplado o es difícil para mantener; mientras que cuestiones relacionadas con el negocio pueden ser el largo tiempo de espera para entregar una nueva funcionalidad o baja productividad de los desarrolladores (Di Francesco, 2018). En consecuencia, adoptar un estilo de Microservicios no es un proceso trivial ya sea para desarrollar un sistema *greenfield*, o bien para migrar un sistema legado que presenta cuestiones difíciles de resolver con el paso del tiempo (Newman, 2015). En algunos casos, migrar hacia microservicios representa una buena opción para resolver los problemas existentes del sistema y, al mismo tiempo, mejorar su mantenimiento y la frecuencia de entregas de funcionalidad.

Los microservicios están contruidos sobre la idea de descomponer un gran servicio en un conjunto de servicios dueños de una sola responsabilidad, los cuales se comunican a través de la red. En este proceso, los servicios pueden escalar eficientemente y los ciclos de entregas se acortan debido al continuo despliegue (Ahmadvand, 2016).

Normalmente, estas decisiones de diseño quedan sujetas a la intuición de desarrolladores y/o arquitectos, pero carecen de un análisis sistemático que les facilite la evaluación de alternativas y toma de decisiones. Se han reportado experiencias interesantes de asistencia inteligente a usuarios en la derivación de diseños orientados a objetos a partir de especificaciones de arquitecturas de software orientadas a servicios (Rodríguez, 2018). En consecuencia, las técnicas de asistencia basadas en *machine learning* podrían servir para sistematizar y facilitar la exploración de diferentes alternativas descomposición de arquitecturas de software en microservicios. Para ello, es necesario contar con una base de conocimiento de aspectos estructurales, funcionales y no funcionales de la arquitectura, que evolucione a largo del tiempo y permita mejorar la precisión de las técnicas definidas y aumente la inteligencia del enfoque propuesto.

En este contexto, el objetivo del proyecto I+D es explorar técnicas de machine learning para asistir a los desarrolladores en descomponer (semi-) automáticamente arquitecturas de software en microservicios, considerando tanto aspectos funcionales como no funcionales.

2. LÍNEAS DE INVESTIGACIÓN, DESARROLLO E INNOVACIÓN

Este proyecto propone un enfoque para migración de arquitecturas monolíticas hacia microservicios y contenedores. El estilo de arquitectura de microservicios está aumentando rápidamente ya que tiene muchas ventajas sobre otros estilos arquitectónicos como la escalabilidad, aislamiento mejorado de fallas (y, por lo tanto, resiliencia) y rendimiento mejorado.

La hipótesis de esta investigación es que existen técnicas de machine learning para detectar microservicios en arquitectura de software monolíticas. De esta manera, es posible identificar relaciones entre las operaciones de un sistema y los estados que toman las diferentes variables variables que

esas operaciones leen o escriben. Entonces es posible visualizar las relaciones entre las operaciones del sistema y las variables reconociendo clusters de relaciones. Asimismo, los aspectos no funcionales contenidos en las especificaciones de microservicios podrían proveer indicios sobre cómo esos microservicios son desplegados en contenedores.

Para corroborar nuestra hipótesis, se proponen los siguientes 4 ejes temáticos:

- Exploración de diferentes alternativas descomposición de arquitecturas de software;
- Visualización de software como asistencia a la descomposición de arquitecturas de software;
- Análisis de especificaciones de APIs REST de aplicaciones basadas en microservicios;
- Integración de arquitecturas de microservicios y contenedores a partir del conocimiento extraído de las especificaciones de los microservicios.

3. RESULTADOS OBTENIDOS/ESPERADOS

El objetivo de este proyecto es desarrollar un enfoque inteligente que asista a los desarrolladores en descomponer (semi-) automáticamente arquitecturas de software en microservicios, considerando tanto aspectos funcionales como no funcionales que contengan indicios sobre cómo se realizará el despliegue de dichos microservicios en contenedores.

Para lograr los objetivos generales descritos anteriormente, se buscará cumplir con un conjunto de objetivos específicos que describimos a continuación. El primer objetivo específico es definir técnicas de machine learning que permitan asistir a los desarrolladores en la exploración de diferentes alternativas descomposición de arquitecturas de software. Para llevar adelante este objetivo es necesario diseñar una metodología que permita vincular las fases de la migración de una arquitectura hacia microservicios. Esta metodología facilitará a

los arquitectos (semi-) automáticamente extraer, transformar, y re-diseñar la arquitectura conservando propiedades claves. Finalmente, para concluir el objetivo se apuntará a construir una base de conocimiento que permita capturar, sistematizar, recuperar conocimiento arquitectónico para evaluar alternativas de descomposición en microservicios. En relación a este objetivo, hemos hallado evidencias en el rol que juegan las técnicas de machine learning en el desarrollo de software orientado a servicios:

- G. Rodríguez, Á. Soria and M. Campo, “AI-based Web Service Composition: A Review” IETE Technical Review, Taylor & Francis, 2015, ISSN: 0256-4602. In Press.
- Rodríguez, G., Soria, Á., Teyseyre, A., Berdun, L., & Campo, M. (2016, September). Unsupervised learning for detecting refactoring opportunities in service-oriented applications. In International Conference on Database and Expert Systems Applications (pp. 335-342). Springer, Cham.
- Rodríguez, G., Soria, Á., & Campo, M. (2016). Artificial intelligence in service-oriented software design. *Engineering Applications of Artificial Intelligence*, 53, 86-104.
- Rodríguez, G., Díaz-Pace, J. A., & Soria, Á. (2018). A Case-based Reasoning Approach to Reuse Quality-driven Designs in Service-Oriented Architectures. *Information Systems*.

El segundo objetivo específico es definir técnicas de machine learning y procesamiento de lenguaje natural para extraer información de aspectos funcionales y no-funcionales a partir de especificaciones de microservicios. Para ello, se creará un meta-modelo que sea capaz de estandarizar las especificaciones de microservicios y tecnología de contenedores con el fin de facilitar la extracción y sistematización de conocimiento. Finalmente, a dicho meta-modelo se le incorporarán aspectos no funcionales, como atributos de calidad y restricciones de despliegue, que puedan proveer soporte para un despliegue

(semi)-automático de los microservicios. En relación a este objetivo, hemos hallado evidencias en el rol que juegan las técnicas de machine learning para análisis de datos, texto y captura sistemática de conocimiento:

- Scott, E., Rodríguez, G., Soria, Á., & Campo, M. (2014). Are learning styles useful indicators to discover how students use Scrum for the first time?. *Computers in Human Behavior*, 36, 56-64.
- Scott, E., Rodríguez, G., Soria, Á., & Campo, M. (2016). Towards better Scrum learning using learning styles. *Journal of Systems and Software*, 111, 242-253.
- Rodríguez, G., Armentano, M., Soria, Á., & Corengia, E. (2020). Evaluation of Markov Models for Architecture Conformance Checking. *IEEE Latin America Transactions*, 18(01), 43-50.
- Vallejos, S., Araujo, L. D. R., Rodríguez, G., Berdun, L., & Toscani, R. (2021). Planificación de IA basada en preferencias para la composición de servicios web. *IEEE Latin America Transactions*, 100(1e).
- Araujo, L. D. R., Rodríguez, G., Vidal, S., Marcos, C. & dos Santos Pereira, R. (2021). An Empirical Analysis on OpenAPI Topic Exploration and Discovery to Support the Developer Community. *Science of Computer Programming*. Elsevier (En evaluación).

Los antecedentes del grupo están vinculados con las investigaciones personales que se han desarrollado vinculadas a la temática propuesta.

El Dr. Guillermo Rodríguez es Investigador Adjunto de CONICET y ha participado en numerosos artículos relacionados a las técnicas de Machine Learning, Educación en Ingeniería de Software, Métodos Ágiles y Arquitecturas Orientadas a Servicios. Es docente de la UNICEN y además se desempeña como Prof. Asociado en UADE y Prof. Titular en CAECE.

El Dr. Fabio Rocha es Profesor Adjunto de la Universidad Tirantes, Scrum Master certificado e investigador con vasta

experiencia en Ingeniería de Software y Microservicios. Es Master en Ciencias de la Computación por la Universidade Federal de Sergipe, y Doctor en Educación por la Universidade Tiradentes. Es coordinador del Laboratorio de Computación Avanzada e Inteligencia Artificial – ITP.

El Dr. Dos Santos es Profesor Asistente en el Departamento de Informática Aplicada de UNIRIO y Jefe del Laboratorio de Ingeniería de Sistemas Complejos (Lab ESC), liderando un equipo de 20 estudiantes. Sus intereses son Ingeniería de Sistemas Complejos (especialmente ecosistemas de software y sistemas de sistemas) y Educación en Ingeniería de Software.

La Dra. Marcos es Doctora en Ciencias de la Computación por la UNCPBA y es Profesora Titular de la UNCPBA con amplia experiencia en Ingeniería de Software y Métodos Ágiles para el Desarrollo de Software.

El Dr. Vidal es docente e investigador de la UNCPBA especializado en evolución y mantenimiento del software. Es Master en Ingeniería de Sistemas y Doctor en Ciencias de la Computación por la UNCPBA.

4. FORMACIÓN DE RECURSOS HUMANOS

La estructura del equipo de trabajo se muestra en la siguiente tabla:

Apellido y nombre	Título	Cargo	Funciones
Rodriguez, Guillermo	Dr.	Prof. UNCPBA	Director
Vidal, Santiago	Dr.	Prof. UNCPBA	Integrante
Teyseyre, Alfredo	Dr.	Prof. UNCPBA	Integrante
Da Rocha Araujo, Leonardo	Lic.	Becario CONICET	Integrante
Catarina, Osmar Felix	Lic.	Becario CONICET	Colaborador
Marcos, Claudia	Dra.	Prof. UNCPBA	Colaborador
Rocha, Fabio	Dr.	Prof. UNIT	Colaborador
Pereira dos	Dr.	Prof.	Colaborador

Santos, Rodrigo		UNIRIO	
Mateos, Crisitan	Dr.	Prof. UNCPBA	Colaborador

El proyecto contempla la participación docente de Facultades de Ciencias Exactas de la UNCPBA, la Universidad Tiradentes, y la UNIRIO; y alumnos de las tres. Por otro lado, se encuentran en desarrollo las siguientes tesis de posgrado:

- Tesis de Doctorado en Ciencias de la Computación. El Lic. Da Rocha Araujo es becario de CONICET y doctorando del doctorado de Ciencias de la Computación de la Facultad de Ciencias Exactas-UNCPBA. Dirección: Guillermo Rodríguez y Rodrigo Pereira dos Santos.
- Tesis de Doctorado en Ciencias de la Computación. El Lic. Catarina es becario de CONICET y futuro doctorando del doctorado de Ciencias de la Computación de la Facultad de Ciencias Exactas-UNCPBA. Dirección: Guillermo Rodríguez y Santiago Vidal.

En relación al presente proyecto se han finalizado las siguientes tesis de grado en Ciencias de la Computación, y Sistemas de Información:

- Um Survey sobre experiências industriais na utilização de Microserviços com Contêineres: Bárbara, Dawitt y Azevedo, Igor. Director: Dr. Fabio Rocha. Finalizado 2020.
- Microservice Architecture: Migrating A Point-of-Sale Application: Dos Santos Moura, Alex. Director: Dr. Fabio Rocha. Finalizado 2020.
- A Novel Unsupervised Learning Approach for Assessing Web Services Refactoring: Hamme, Brian y Listorti, Luciano. Director: Dr. Guillermo Rodríguez. Co-director: Dr. Cristian Mateos. Finalizado 2019.

5. BIBLIOGRAFÍA

- Ahmadvand, M., & Ibrahim, A. (2016). Requirements reconciliation for scalable and

secure microservice (de) composition. IEEE International Requirements Engineering Conference Workshops.

- Di Francesco, P., Malavolta, I., & Lago, P. (2018). Migrating towards Microservice Architectures: an Industrial Survey. IEEE International Conference on Software Architecture, ICSA 2018, Seattle, USA.

- Fowler, M. & Lewis, J. (2018). Microservices a definition of this new architectural term. URL: <http://martinfowler.com/articles/microservices.html>

- Newman, S. (2015). Building microservices: designing fine-grained systems. O'Reilly Media, Inc.

-Rahman, M. and Gao, L. (2015). A reusable automated acceptance testing architecture for microservices in behavior-driven development. IEEE Symposium on Service-Oriented System Engineering (SOSE).

- Rodríguez, G., Díaz-Pace, J. A., & Soria, Á. (2018). A Case-based Reasoning Approach to Reuse Quality-driven Designs in Service-Oriented Architectures. Information Systems.