Uso de Wrapper y Cloud Computing para la generación de servicios a partir de aplicaciones heredadas

Gustavo López

Instituto de Ingenierías y Nuevas Tecnologías de la Universidad Nacional del Oeste
gustavolopez@uno.edu.ar

Enrique Fernández

Instituto de Ingenierías y Nuevas Tecnologías de la Universidad Nacional del Oeste
enfernandez@uno.edu.ar

Eugenia C. Artola

Universidad Nacional de Cuyo
ecartola@hotmail.com

Resumen

Uno de los procesos de modernización de un sistema heredado, implica combinar procesos de ingeniería inversa e ingeniería directa. La ingeniería inversa consiste en partir desde un sistema, detectar sus módulos y relaciones, para luego modelar el mismo en otro con un mayor grado de abstracción.

Partiendo de la definición de estos modelos se aplican transformaciones que utilizan la ingeniería directa. De allí la combinación de ingeniería inversa y directa. En esta combinación se pueden utilizar procesos automatizados en mayor o menor medida con tecnología basada en la nube.

Los sistemas heredados no resultan fáciles de mantener ni de sustituir, es por ello que en esta propuesta se involucra el uso del paradigma Model Driven Engineering (MDE) y se contempla la modernización por aplicación de ingeniería inversa, que comienza en niveles bajos de abstracción utilizando las plataformas tecnológicas origen del proceso, para luego aplicar ingeniería directa, que permite transformar modelos de alto nivel de abstracción en modelos de menor nivel, hasta llegar al código del nuevo sistema destino.

Se contempla el proceso de modernización de los estándares propuestos por el Grupo de trabajo de modernización (ADM), impulsada por la arquitectura del consorcio Object Management Group (OMG) y las distintas contribuciones que han extendido estos estándares.

Palabras Claves: *Sistema heredado. Ingeniería directa. Ingeniería inversa. Model Driven Engineering. Cloud computing.*

Contexto

La presente línea de investigación forma parte del proyecto de Investigación y Desarrollo identificado como PID AC 05/22 “Propuesta de una PaaS con generación automática de servicios por exposición de aplicaciones legacy para ser consumidos por clientes SaaS. Definición y diseño de una plataforma que se sustente en los conceptos de SaaS, IaaS y Cloud Computing” aprobado y acreditado por D. SCyT 04/2023 de la Universidad Nacional del Oeste.

Introducción

Un proceso de modernización de software implica cambios sustanciales en un sistema informático existente, denominado sistema de origen; el cual no es un proceso de mantenimiento del software [2]. Las técnicas empleadas son el metamodelado y las transformaciones entre modelos [3]. En Favre [1] se sientan los fundamentos de la modernización de software dirigida por modelos Model Driven Software Modernization (MDSM), basado en el paradigma Model Driven Engineering (MDE).

La reingeniería de un sistema es un tipo de modernización, cuyo fin es mejorar la calidad del sistema [4, 5], manteniendo el valor estratégico del sistema existente [6].

Un proceso de **modernización** consta de tres etapas [7]:

1) Ingeniería inversa;
2) Reestructuración;
3) Generación o ingeniería directa.

1) La ingeniería inversa se inicia con la inyección de modelos a partir de artefactos como un código o un esquema relacional del sistema origen.

2) La reestructuración consiste en una sucesión de transformaciones modelo-a-modelo que pueden elevar el nivel de abstracción de cada modelo obtenido en función de la semántica del meta - modelo correspondiente.

3) La generación o ingeniería directa es otra cadena de transformaciones de modelo-a-modelo que van disminuyendo el nivel de abstracción, pudiendo llegar hasta obtener el código final en el sistema destino.

Un proceso de **migración** es un caso especial de reingeniería donde sólo se cambia la plataforma tecnológica, manteniendo la funcionalidad del sistema [8, 9].

El proceso de modernización del software comprende el tratamiento de aplicaciones heredadas, las cuales no son fáciles de mantener ni de sustituir. La mayoría de los métodos, actualmente, están basados en el paradigma MDE y contemplan caminos de modernización como [10, 11]:

- La **ingeniería inversa,** que parte de niveles bajos de abstracción que empiezan con las plataformas tecnológicas que dan origen al proceso: código y bases de datos.
- La **ingeniería directa**, que permite transformar modelos de alto nivel de abstracción en modelos de menor nivel, hasta llegar al código del nuevo sistema destino.

La tecnología en la nube permite la disponibilidad bajo demanda de recursos de computación como servicios a través de Internet. Esta tecnología evita que las empresas tengan que encargarse de aprovisionar, configurar o gestionar los recursos y admite que abonen únicamente por lo que utilicen [14].

Hay tres tipos de modelos de servicio en la nube:

- *infraestructura como servicio*, ofrece servicios de computación y almacenamiento;
- *plataforma como servicio*, proporciona un entorno de desarrollo y despliegue para crear aplicaciones en la nube y,
- *software como servicio*, facilita aplicaciones como servicios [15].

Resultados Obtenidos / Esperados

- Generar servicios de forma automática partiendo de aplicaciones heredadas que se sustente en Cloud computing.
- Definir servicios utilizando la técnica del metamodelado.
- Utilizar técnicas de reingeniería de sistemas a fin de generar servicios.
- Integrar los servicios que se generaron a fin de exponer sistemas heredados.
- Utilizar Cloud computing a efectos de la disponibilidad de los servicios.

Estado actual y evolución

El crecimiento del mercado en diferentes ramas de negocios ha provocado que el desarrollo de las TIC's se aplique en las diversas organizaciones para el logro de sus objetivos [15].

Así mismo, actualmente las TIC's reflejan el modelo de negocio de una organización y proporcionan el apoyo a la funcionalidad de la misma [9].

El deseo de las empresas por crecer y por ser diferentes ante sus competidores obligó a que las TIC's tengan una evolución inmediata. Las empresas, para formar parte del mercado, se dedican a solicitar el desarrollo de aplicaciones basadas en necesidades específicas como liquidación de sueldos o facturación, sin tener en cuenta la infraestructura tecnológica de las aplicaciones que forman parte de su software corporativo [15].

Este hecho ha originado que algunas aplicaciones no puedan interactuar entre sí con la

consecuente demora en la ejecución de los procesos. También produce que exista cierta redundancia de datos, por ejemplo, la Clave Única de Identificación Tributaria (CUIT) de un cliente [3].

El paradigma actual dentro del diseño de arquitecturas, estimula que las aplicaciones de una organización trabajen de forma integrada sin importar bajo qué plataforma están desarrolladas. Es un marco de trabajo que permite a las empresas centrarse en la resolución de sus objetivos sin tener en cuenta las infraestructuras con las que cuentan para llevar a cabo sus negocios [13].

La evolución física se ha centrado en el uso de redes; estas permiten que varias computadoras puedan trabajar compartiendo recursos informáticos. Los primeros programas fueron los Monolíticos y estaban preparados para ejecutarse en una estación de trabajo. Luego se impusieron los desarrollos basados en Cliente-Servidor. Por último, nace Internet, como la herramienta que permite la conexión de millones de computadoras a través de una infinidad de recursos informáticos [12]. De la mano de Internet surgen los Servicios Web como metodología que promueve la Heterogeneidad e Interoperabilidad entre aplicaciones desarrolladas en diferentes plataformas [15].

Los Sistemas Monolíticos tienen vigencia entre los años 1950 y 1980. Fueron desarrollados en lenguajes como Cobol y Fortran. Estos sistemas tienen como característica que todos los módulos del software se ejecuten en la misma computadora donde se encuentra la base de datos a la cual acceden.

Los desarrolladores de estos sistemas previeron que era necesario descomponer el software para evitar la complejidad. En consecuencia, nació la Programación Estructurada dividiendo en módulos todos los componentes del sistema [12].

Alrededor de los años ’80 surgen estos tipos de sistemas. Se definen por poseer las aplicaciones que procesan los datos en una computadora Cliente y la base de datos en otra computadora Servidor. Se los denominó Sistemas Cliente-Servidor. Los mismos se comunican a través de conexiones de red. La separación es tanto lógica como física. La principal dificultad que presentaron este tipo de sistemas se basó en que las capas de las aplicaciones que exponían los datos a los usuarios se inmiscuían con las capas que procesaban a los mismos en la base de datos [2].

Para solucionar la problemática que presentaban los Sistemas Cliente-Servidor, surgieron en los años ’90 los Sistemas N-Tier ó Multicapas. En este tipo de sistema se representó en diferentes capas a la Interfaz del Usuario, a la Lógica del Negocio y al Manejo de Base de Datos [2].

Así, los sistemas heredados fueron, son y probablemente seguirán siendo el corazón de muchas organizaciones que automatizaron sus procesos de negocios hace décadas con tecnologías que actualmente son anticuadas, pero siguen siendo útiles para las empresas.

Hoy en día, el acelerado avance tecnológico, la evolución de los paradigmas en relación a la arquitectura del software y el uso de plataformas, permiten que la gran mayoría de los negocios y organizaciones basen sus procesos en sistemas informáticos mucho más evolucionados, lo que hace imprescindible que estos nuevos procesos convivan con los sistemas heredados [15].

Esta clase de sistemas heredados, como también las organizaciones, deben evolucionar y adaptarse a nuevos requerimientos.

Según Espinel [11] las soluciones clásicas relacionadas son:

- el *rediseño* completo del sistema en una nueva tecnología;
- el *wrapping* del sistema heredado que provee una nueva interfaz para algún componente de éste, lo que permite mayor accesibilidad desde otras aplicaciones;
- la *migración* o transformación que mueve al sistema heredado a un nuevo ambiente o plataforma más flexible, reteniendo la funcionalidad y los datos del sistema original.

Una de las técnicas para migrar sistemas heredados hacia nuevas tecnologías se logra

realizando llamadas a las funcionalidades del sistema heredado existente mediante el uso del Wrapper [9].

Mediante el wrapper, es posible adaptar al sistema heredado, para que sea parte de una nueva generación de sistemas, sin los riesgos inherentes a la reescritura y los altos costos de nuevos desarrollos [9]. Es evidente, que para que esto no introduzca problemas a futuro, el estado de salud del sistema debe ser "sano" o debe ser factible pasarlo a dicho estado.

Wrapper implica una función contenedora del sistema heredado, a modo de subrutina, en una biblioteca de software cuyo propósito principal es llamar al sistema con poco o ningún cálculo adicional [16].

Los estudios de Zaragoza [9], indican que los sistemas heredados son considerados potencialmente problemáticos por numerosos ingenieros de software por diversos motivos. Muchas organizaciones no han logrado los resultados esperados tras migrar, re implementar o integrar sus sistemas heredados.

Muchos de los sistemas de información todavía corren sobre plataformas heredadas.

Entre los muchos ejemplos de sistemas actuales que siguen funcionando bajo plataformas heredadas, se pueden mencionar:

- Sistemas contables.
- Sistemas bancarios.
- Aplicaciones Rapid Application Development (RAD) orientadas a interfaces gráficas.
- Sistemas empresariales basados en procesos de negocios.

Líneas de Investigación y Desarrollo

Un sistema heredado es un sistema, tecnología o aplicación de software antiguo y desactualizado que sigue en uso dentro de una organización porque sigue desempeñando las funciones para las que fue diseñado [12]. Los sistemas heredados ya no cuentan con soporte y mantenimiento y están limitados a nivel de crecimiento [9, 10]. Sin embargo, no pueden reemplazarse fácilmente.

Este escenario casi siempre desemboca en la necesidad de volver a desarrollar nuevamente una parte importante de las aplicaciones heredadas para exponer las mismas en forma de servicios, debiendo reinvertir tiempo y dinero en partes del sistema que ya estaban desarrolladas y probadas.

En este punto es en donde se detecta la necesidad de analizar las estructuras internas y los metadatos de las aplicaciones ejecutables, intentando abrir virtualmente la aplicación, localizar si es que hubiera puntos de extensión que permitan exponer automáticamente sus funcionalidades como servicios, que respeten los estándares de la industria y por lo tanto faciliten la integración con herramientas de terceros [12]. Para ello se evaluará la posibilidad de utilizar tecnología basada en la nube [14].

Se buscará utilizar Infraestructura como servicio en línea a fin de proporcionar un alto-nivel de APIs utilizadas para implementar detalles a bajo nivel de infraestructura como recursos de informática física, ubicación, seguridad, copia de seguridad, etcétera.

La finalidad es lograr que las empresas utilicen aplicaciones e infraestructura basadas en la web sin tener que comprar, administrar y brindar soporte a la infraestructura de nube subyacente [14].

Utilizando una plataforma informática como un servicio se puede reducir la cantidad de tiempo y esfuerzo dedicado al desarrollo, implementación y actualizaciones. Los desarrolladores pueden crear aplicaciones web rápidamente sin instalar ninguna herramienta o software [16].

Formación de Recursos Humanos

El equipo de trabajo está formado por tres doctores y un becario, todos del área de conocimiento a la que se relaciona el proyecto.

Se prevé el trabajo sobre tesis de grado y trabajos profesionales o de fin de carrera llevados a cabo en forma colaborativa entre la Universidad Nacional del Oeste y la Universidad Nacional de Cuyo. El equipo ya ha presentado algunos trabajos relacionados a la presente línea de investigación.

Conclusiones

Reemplazar un sistema heredado es complejo, costoso y requiere tiempo. Por eso es esencial evaluar aspectos clave como el estado del sistema, cuáles son las necesidades del negocio o qué riesgos son asumibles [7, 11]. Asimismo, si el sistema heredado está desarrollado con una tecnología o lenguaje de programación obsoleto, puede ser difícil encontrar profesionales calificados para implementar la migración [9].

Las empresas que en el pasado han invertido una gran cantidad de su presupuesto en el desarrollo de software, tienen la necesidad de implementar el paradigma de servicios y adaptar en consecuencia el software existente [13].

Este cambio, muchas veces no implica la incorporación de requerimientos funcionales nuevos dentro de las aplicaciones existentes, sino que se pretende que las mismas expongan su funcionalidad en forma de servicios permitiendo una integración con otras aplicaciones [9].

Referencias bibliográficas

[1] Favre, J. M. "Foundations of model (driven)(reverse) engineering: Models", Proceedings of the International Seminar on Language Engineering for Model-Driven Software Development, Dagstuhl Seminar 04101. 2004.

[2] Pressman, R.S. "Ingeniería del software. Un enfoque práctico". Mcgraw-Hill, 7ª Ed, ISBN: 978-607-15-0314-5. 2010

[3] Almonte, L., Guerra, E., Cantador, I. "Recommender systems in model-driven engineering. Software Syst Model 21, 249–280. https://doi.org/10.1007/s10270-021-00905-x. 2022.

[4] Tilley, S. R., Smith, D. "Perspectives on legacy system reengineering. Technical report". Software Engineering Institute, Carnegie Mellon University. 1995.

[5] Müller, H. A., Jahnke, J. H., Smith, D. B., Storey, M. A., Tilley, S. R., & Wong, K. "Reverse engineering: A roadmap". In Proceedings of the Conference on the Future of Software Engineering (pp. 47-60). ACM. 2000.

[6] Castillo, R., Guzmán, I.. "Knowledge Discovery Metamodel-ISO/IEC 19506: A standard to modernize heredado systems. Research Gate". University of Castilla-La Mancha 2021. Web. 25 Jul. 22. 2021.

[7] Hunold, S., Korch, M. "Transformation of Legacy Software into Client/Server Applications through Pattern-Based Rearchitecturing". Research Gate. Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International.2008.Web. 15 May. 2022.

[8] Holley, K., Channabasavaiah, K.. "Migrating to a service-oriented architecture". IBM Gobal Services. IBM Corporation Software Group 2004. Web. 5 Jul. 2016.

[9] Zaragoza, P., Seriai, AD., Seriai, A., Shatnawi, A., Bouziane, HL., Derras, M.. "Materializing Microservice-oriented Architecture from Monolithic Object-oriented Source Code". In: Fill, HG., van Sinderen, M., Maciaszek, L.A. (eds) Software Technologies. ICSOFT 2021. Communications in Computer and Information Science, vol 1622. Springer, Cham. https://doi.org/10.1007/978-3-031-11513-4_7. 2022

[10] Salvatierra, G, Zunino, A.. "Towards a Computer Assisted Approach for Migrating Legacy Systems to SOA". Research Gate. International conference on Computational Science and Its Applications 2012. 2012. Web. 10 May. 2022.

[11] Espinel, G. P., Carrillo Medina, J. L., Flores Calero, M. J., Urbieta, M. "Software Configuration Management in Software Product Lines: Results of a Systematic Mapping Study". IEEE Latin America Transactions. vol. 20, no. 5, pp. 718-730, May 2022, doi: 10.1109/TLA.2022.9693556.

[12] Fanomezana, M. L., Rapatsalahy, A. M., Razafindrakoto, N. R., Bădică, C. "Proposed Methodology for Designing a Microservice Architecture". 23rd International Carpathian Control Conference (ICCC), 2022, pp. 303-308, doi: 10.1109/ICCC54292.2022.9805930.

[13] Monteiro, A., Vieira, G. Guiding legacy systems for evolution. PmatE: a case study of maintenance and engineering. Journal of Information Systems Engineering and Management,7(1),11689. 2022. https://doi.org/10.55267/iadt.07.11689

[14] IBM SOA. "Integrate legacy systems into your SOA initiative". Web. 4 Jul 2022. <https://www.ibm.com/developerworks/library/ws-soa-legacyapps/>.

[15] Microsoft Web ServiceS. Web. 4 Jul 2022. <https://msdn.microsoft.com/en-us/library/ms950421.aspx>.

[16] Grechanik, M., Conroy, K. "Composing Integrated Systems Using GUI-Based Applications And Web Services", Web Services (ICWS) IEEE International Conference on, pp., 2007. Web. 30 May. 2022.