

Desarrollo de aplicaciones eHealth basadas en microservicios en una arquitectura de Cloud

Hurviades Calderón-Gómez¹, Francisco Navarro-Marín¹, José Manuel Gómez-Pulido¹, José Luis Castillo-Sequera¹, Alberto Garcés-Jiménez², María-Luz Polo-Luque³, José Sanz-Moreno³, Gloria Sención⁴, Miguel Vargas-Lombardo⁵

hurviades.calderon@edu.uah.es, francisco.navarro@edu.uah.es, jose.gomez@uah.es, jluis.castillo@uah.es, albertogarceso@gmail.com, mariluz.polo@uah.es, josesanzm@gmail.com, dragloriasencion@gmail.com, miguel.vargas@utp.ac.pa

¹ Health Computing and Intelligent Systems Research Group (HCIS), Universidad de Alcalá, Alcalá de Henares, 28801, Madrid, España.

² Foundation for Biomedical Research of Príncipe de Asturias University Hospital (FIBHUPA), Universidad de Alcalá, Alcalá de Henares, 28801, Madrid, España.

³ Departamento de Enfermería y Fisioterapia, Universidad de Alcalá, Alcalá de Henares, 28801, Madrid, España.

⁴ Escuela de Medicina, Universidad Autónoma de Santo Domingo (UASD), Santo Domingo, 10103, Santo Domingo, República Dominicana.

⁵ Grupo de Investigación en Salud Electrónica y Supercomputación (GISES), Universidad Tecnológica de Panamá, Panamá, 0819-07289, Panamá, Panamá.

Pages: 81–93

Resumen: En este trabajo se describe una propuesta orientada a los emergentes servicios de eHealth, para ofrecer una mejor prestación en la atención asistencial y hospitalaria. El enfoque propuesto se apoya en el desarrollo de microservicios consistentes en proporcionar una aplicación que contiene a su vez una serie de pequeños y adaptables servicios que se ejecutan de forma autónoma, comunicándose de forma interactiva y continua. Así, se pretende fortalecer los sistemas de salud tradicionales en los que todavía se carece del Registro Médico Electrónico (RME) y los sistemas de información disponibles presentan una escasa disponibilidad en el acceso y en el procesamiento de sus datos, lo que limita la toma de decisiones clínico-médicas. Se presentan dos sistemas de información desarrollados con este enfoque, con arquitecturas de software que presentan capas estables y modificables, adaptadas a las necesidades de cualquier organización, escalables y replicadas en varios contenedores la infrutilización de recursos informáticos.

Palabras-clave: Microservicios; eHealth; SaaS; Arquitectura de Software; Data Clínica

Development of eHealth Applications-based on microservices in a cloud architecture

Abstract: This paper describes a proposal aimed at the emerging services of eHealth, to offer a better provision of personal care and hospital care. The proposed approach

is based on the development of microservices consisting in providing an application that contains a series of small and adaptable services that are executed autonomously, communicating interactively and continuing. With this, it is intended to strengthen traditional health systems in which the Electronic Medical Record (EMR) is still lacking and the information systems has limited availability in the access and processing of their data, which limit decision-making in the clinical-medical environment. Two information systems developed with this approach are presented, with software architectures that have stable and modifiable layers, adapted to the needs of any organization, scalable and replicated in several containers without underutilizing computer resources.

Keywords: Microservices; eHealth; SaaS; Software Architecture; Medical History

1. Introducción

El presente trabajo presenta una arquitectura software basada en capas y microservicios como estrategias software en el contexto de la eHealth. Dichas estrategias deberán potenciar y facilitar mejoras sustanciales en la calidad de servicios TIC's en salud; reduciendo costos, mejorando la igualdad en los accesos a la información clínica vía internet, y en la autonomía de los profesionales en la asistencia clínica (médicos, enfermeras y auxiliares de enfermería). La aplicación de los microservicios, basada en las arquitecturas de software en capas, ha sido utilizada previamente en la construcción de Sistemas de Información Clínicos como es el caso de la Clínica Universitaria de la Universidad Tecnológica de Panamá, con el objetivo de consolidar los procesos de prestación de servicios eHealth, realizando una reingeniería de los procesos clínicos simplificando y mejorando el apoyo a las tareas de diagnóstico del especialista médico. Como indican (Gholap, Janeja, & Yesha, 2015; Vilaplana, Solsona, Abella, Filgueira, & Rius, 2013), estos desarrollos de software deberán orientar y mejorar el impacto en los servicios *eHealth* optimizando la comunicación entre sistemas de salud —un sistema de salud comprende todas las organizaciones, instituciones, recursos y personas cuya finalidad primordial es mejorar la salud (Organización Mundial de la Salud, 2019). Por otro lado, el proyecto “*Diseño e implementación de un sistema inteligente de bajo coste para el prediagnóstico y la teleasistencia de enfermedades infecciosas en personas de edad avanzada SPIDEP*” ERANET LAC 2015, con el objetivo de construir un marco trabajo basado en las TIC para apoyar el diagnóstico temprano de enfermedades infecciosas. El proyecto incluye el diseño, desarrollo e implementación de una plataforma de Hardware capaz de soportar la comunicación del sistema generalizado con el entorno de la nube central (Nube en Salud). Además de un subsistema capaz de apoyar el diagnóstico clínico asistido en el campo de las enfermedades infecciosas. Este subsistema incluirá un sistema de inferencia basado en Aprendizaje automático para mejorar el soporte de decisiones para la prevención, el tratamiento y el manejo de enfermedades infecciosas.

Consideramos evidente la interdisciplinariedad en el que se halla inserta la eHealth al pertenecer a un extenso dominio al que se le aporta experiencia, esfuerzos, consensos y capacidades de diversas áreas como: interacción persona-computadora, ingeniería del software, redes sociales, telecomunicaciones, ciencias sociales, medicina, enfermería, salud pública, electrónica, robótica, etc. Actualmente, existen muchos aspectos que se deben considerar sobre el concepto *eHealth*, debido a esto, hemos decidido delimitarlo a cinco secciones para esta investigación para así describir la importancia del tema y las razones que justifican su estudio. La sección 1 aborda la presente introducción en la que

se describe la importancia del tema y las razones que justifican su estudio. En la sección 2, se examina el uso de microservicios en el contexto de eHealth; en la 3, se describen las etapas para el desarrollo de las arquitecturas de softwares en capas orientados a los microservicios; en la 4, se muestran los resultados obtenidos por las diversas pruebas de *softwares*; y en la sección 5, se vinculan los resultados con la discusión. Por último, compartimos la conclusión y exponemos la contribución de esta investigación.

2. Estado del arte: Arquitectura basada en microservicios

Recientemente, este enfoque arquitectónico ha recibido atención significativa desde el punto de vista de los investigadores en las diferentes áreas del conocimiento, como, por ejemplo: system quality, smart city clouds, migration or mobile-oriented (Di Francesco, Lago, & Malavolta, 2019; Götz et al., 2018; Krämer, Frese, & Kuijper, 2019; Li et al., 2019); sin embargo, hay pocas investigaciones enfocada al área de eHealth, específicamente al diagnóstico clínico asistido en enfermedades infecciosas, a comparación con otros trabajos relacionados (Esposito, Castiglione, Tudorica, & Pop, 2017; Jita & Pieterse, 2019; Raković & Lutovac, 2015).

Este estudio aporta una nueva visión de lo ya expuesto en las investigaciones previamente citadas, ya que nuestra contribución está orientada al diseño de las arquitecturas de capas basada en microservicios para el diagnóstico clínico asistido en el campo de las enfermedades infecciosas, facilitando así el control regular de las constantes vitales a distancia (telemedicina) de los pacientes por parte del personal médico. En este sentido, una atención preventiva minimiza la severidad de los procesos infecciosos y, colateralmente, la reducción de recursos necesarios para controlar adecuadamente el problema.

Entre la serie de ventajas que proporciona aplicar este enfoque arquitectónico se encuentra; en primer lugar, cada componente se puede duplicar para equilibrar la carga, según sea necesario (Di Francesco et al., 2019); en segundo lugar, los microservicios al aplicar una escala funcionalidad pequeña, permite que la reacción a los picos de demanda sea más rápida y precisa (Elsayed & Zulkernine, 2019), y por último al aplicar el concepto de modularidad los componentes hace que el sistema sea robusto a los fallos (Götz et al., 2018). Sin embargo, al aplicar este enfoque hay que tener presente las siguientes dificultades: en primer lugar, cada solicitud forzaría al RestAPI a transporta los datos de un servicio a otro. Esto conduce a una mayor redundancia y la integridad de los datos ya no está garantizada. Por lo tanto, este método solo se recomienda si los datos no tienen que estar actualizados para cada solicitud (Yogesh, Pushkara, Shilpa, Malati, & Rashmi, 2018); en segundo lugar, se requiere del diseño de una arquitectura distribuida enfocada a los desafíos que conlleva, como, por ejemplo: latencia, falta de fiabilidad de la red, la tolerancia a fallos, la organización de servicios complejos, la coherencia de los datos y la gestión de transacciones, y el equilibrio de carga. En consecuencia, las infraestructuras en la nube desempeñan un papel fundamental para el funcionamiento de los microservicios y su complejidad (Abdullah, Iqbal, & Erradi, 2019; Di Francesco et al., 2019).

3. Métodos y Materiales: Microservicios en SaaS

Para la realización de esta arquitectura, se ha optado por el paradigma de la computación en la nube (SaaS), orientado a la implementación de microservicios, cuyo propósito es

ofrecer acceso a una serie de pequeños servicios de *software* (Newman, 2015) para la gestión y consolidación de la información clínica de los pacientes y no solo lo necesario para la generación de un reporte estándar, sino que también influye en la interoperabilidad y en la estructura de datos con otros sistemas de información médicos (Catley & Frize, 2003; Handayani et al., 2017). Por ello, la jerarquización de la información y sus datos deberán ser contemplados como un todo que fluye hacia la organización y sin sesgos entre departamentos de la organización, con el objetivo de apoyar a la toma de decisiones con indicadores parametrizados, consistentes y veraces de los datos recopilados.

Por tanto, basándonos en diversas investigaciones (Cho & Chan, 2015; Mousa, Bentahar, & Alam, 2019; Oliveira, Martins, Sarker, Thomas, & Popović, 2019; Sharma, Kumar, & Agarwal, 2015), se identificó que el modelo SaaS posee una serie de características, en comparación con los modelos IaaS, PaaS y XaaS, que lo hacen particularmente interesante para el contexto *eHealth*: (I) bajo costo de implementación; (II) mejora la escalabilidad y accesibilidad, así como la facilidad de administración y la eficiencia operativa en comparación con los sistemas monolíticos (Abdullah et al., 2019); (III) los usuarios finales ejecutan sus aplicaciones sin tener ningún control sobre el entorno host (Elsayed & Zulkernine, 2019; Stavrinides & Karatza, 2019); (IV) cada usuario tiene una instancia independiente de un servicio SaaS, ya que se encuentra virtualizados y heterogéneos '*Docker Host*' (El Kafhali & Salah, 2018); (V) mejora la interoperabilidad y la integración a través de un conjunto de servicios web para proporcionar funcionalidades personalizadas a los usuarios (Fan, Hussain, Younas, & Hussain, 2015).

Dentro de este marco, se desarrolla e implementa el uso de microservicios en dos proyectos de investigación enfocado al contexto de eHealth (Clínica Universitaria y SPIDEP) que se describen en los siguientes sub-secciones.

3.1. Clínica Universitaria de la Universidad Tecnológica de Panamá

Entre 2015 y 2017, se desarrolló el proyecto Clínica Universitaria, el cual se basa en las estrategias y en el plan de acción sobre *eHealth* (Organización Panamericana de la Salud, 2011), con el propósito de mejorar la calidad de atención de los profesionales de la salud. Para este estudio, se propuso desarrollar un plan estructurado y semántico en beneficio del bienestar y la salud de toda la comunidad académica de la Universidad Tecnológica de Panamá (UTP), campus central, siete centros regionales y dos extensiones universitarias. Con el objeto de registrar las afecciones más comunes que aquejan a los distintos estamentos (estudiantes, docentes, investigadores y administrativos) de la institución y generar indicadores a partir de los datos generados mediante el formato CIE-10.

Como complemento, se decidió que el mejor método para el diseño centrado en el usuario (DCU) de la clínica universitaria, sería la metodología 'Persona' de Alan Cooper (Cooper & Reimann, 2004). Esta técnica se llevó a cabo bajo los modelos usuario y dominio: el primer modelo se enfoca en cómo el usuario final (profesionales de la salud) interactúa con el sistema para lograr sus tareas, características y metas. Además, obtener un diseño usable y escalable de los servicios; el segundo modelo se orienta a optimizar los diferentes procesos involucrados con el usuario con aplicaciones de terceros.

Como seguimiento de esta actividad, se desarrolló un diseño de una arquitectura de software (3 capas), aplicado a la clínica universitaria bajo una versión 'candidata a definitiva (RC)'. En la figura 1 muestra cómo interactúan las capas y sus componentes.

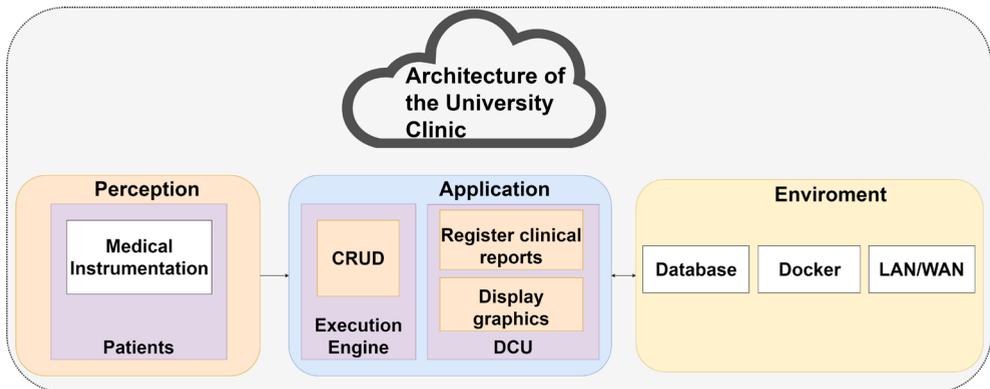


Figura 1 – Diseño de una arquitectura de software de tres capas, aplicado a la clínica universitaria UTP

En tal sentido, se debe tener presente que la arquitectura de capas variaría dependiendo de los requisitos funcionales y no funcionales de los usuarios. En consecuencia, existen diferentes investigaciones que han propuesto distintos niveles de capas dentro de la arquitectura (Kumar & Mallick, 2018; Saadeh, Sleit, Sabri, & Almobaideen, 2018; Sharma et al., 2015). A continuación, se ofrece una descripción general de nuestro esquema modificado de la arquitectura de tres capas, enfocado al desarrollo de aplicaciones intrauniversitario o intraorganizacional:

- Percepción (Perception): Esta primera capa permite la recopilación global de los datos médicos de los pacientes de la UTP. Dichos datos son recolectados por los profesionales de salud mediante el uso de instrumentos médicos como básculas, glucómetros, esfigmomanómetros (Kumar & Mallick, 2018).
- Aplicación (Application): Esta segunda capa administra los dos grupos de microservicios exclusivos para los profesionales de la salud y los administradores de cada región universitaria con sus respectivos privilegios en la nube, con el objetivo de permitir el acceso a distintas funcionalidades como visualizar la historia clínica del paciente, incorporar nuevos datos registrados, gestionar el personal médicos y sus roles; es decir, su característica principal es ofrecer un conjunto de microservicios específicos, según el tipo de aplicación para el usuario final (Ciavotta, Alge, Menato, Rovere, & Pedrazzoli, 2017; Kiss et al., 2019).
- Entorno (Environment): Esta tercera capa soporta el almacenamiento y control de los datos a través de las bases de datos distribuidas. Además, esta capa gestiona la intercomunicación de los datos entre los dos grupos de microservicios (capa de aplicación) y los instrumentos médicos de las diferentes regiones universitarias (capa de percepción), mediante el uso de la red LAN/WAN (Saadeh et al., 2018).

3.2. Diseño e implementación de un sistema inteligente de bajo coste para el prediagnóstico y la teleasistencia de enfermedades infecciosas en personas de edad avanzada (SPIDEP)

Se propone un modelo SaaS para apoyar en el diagnóstico de enfermedades infecciosas de pacientes (adultos mayores) en residencias médicas (centros médicos de Santo Domingo,

en la República Dominicana, y de las residencias de la Consejería de Políticas Sociales-Familia de la Comunidad de Madrid, España), mediante el uso de registros de parámetros clínicos definidos (tensión arterial, frecuencia de pulso, temperatura corporal, y entre otros), obtenidos a través de sensores biométricos no invasivo (Fiedler, 2017). Se hace especial énfasis en el diseño, desarrollo, implementación e interoperabilidad de una plataforma de bajo coste, capaz de soportar la comunicación del servicio generalizado con el entorno de la nube central. Asimismo, se incluirá un subsistema de aprendizaje automático para mejorar el soporte de decisiones para la prevención, el tratamiento y el manejo de enfermedades infecciosas. Sobre la validez de la experiencia de usuario (UX), fue aprobado por el personal de salud en conjunto con los investigadores del consorcio internacional mediante el uso de *mock-up* (Alharbi, Atkins, & Stanier, 2017; Leung & Chen, 2019). Por otro lado, el funcionamiento técnico de la nube cumple con los estándares establecidos a través de las diferentes pruebas de *softwares*.

Aunado a lo anterior, se ha llevado a cabo una propuesta preliminar de la arquitectura de *software* (5 capas), aplicado a SPIDEP bajo una versión 'Beta'. En la figura 2 muestra cómo interactúan las capas y sus componentes.

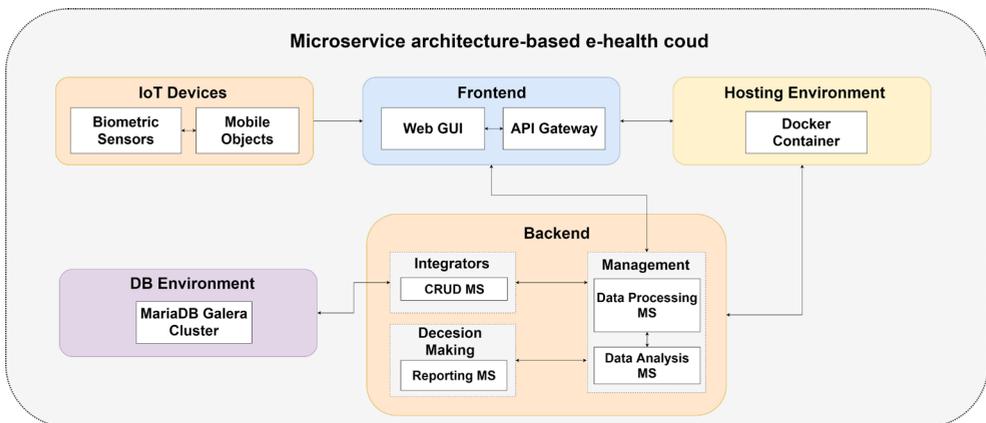


Figura 2 – Diseño de una arquitectura de software de cinco capas, aplicado a SPIDEP

De esta manera, se ofrece una descripción general de nuestro esquema modificado de la arquitectura de cinco capas, enfocado al desarrollo de aplicaciones interuniversitarias o interorganizacionales:

- **Dispositivos IoT (*IoT Devices*):** Esta primera capa permite una interacción entre la capa Frontend y la capa Backend por medio de elementos sensores biométricos WBAN no invasivos (Raković & Lutovac, 2015), para la comprobación de los parámetros clínicos de los pacientes (tensión arterial, frecuencia de pulso, temperatura corporal y entre otros), que son obtenidos a través de un dispositivo móvil.
- **Frontend:** Esta segunda capa muestra la información sobre el estado de los pacientes, bioseñales relacionadas y el contenido de los datos obtenidos de la capa de dispositivos IoT mediante una interfaz *web* (GUI). Además, esta capa

emplea el uso de *API Gateway* (Dinh, Lee, Niyato, & Wang, 2013), para la intercomunicación con la capa de *Backend*.

- *Backend*: Esta tercera capa gestiona a nivel lógico todos los grupos de microservicios creados para los profesionales de la salud, los administradores de cada residencia y los colaboradores del proyecto SPIDEP con sus respectivos privilegios en la nube, para permitir acceder a las distintas funcionalidades, como visualizar la historia clínica del paciente; recuperar, modificar y cargar los datos médicos; gestionar roles y usuarios; permitir el uso de un sistema de prediagnóstico para recomendar directrices a seguir por parte del sistema de inferencia (Toma de Decisiones) y, entre otros microservicios (Abdullah et al., 2019).
- Entorno de Base de datos (DB Environment): Esta cuarta capa administra el almacenamiento y control de los datos a través de las bases de datos distribuidas. Además, permite el acceso de los clientes en la nube mediante el uso de consultas predefinidas (Furda, Fidge, Barros, & Zimmermann, 2017).
- Entorno del *Host (Hosting Environment)*: Esta quinta capa soporta la disponibilidad y seguridad de datos con sus respectivos componentes entre las capas superiores; es decir, incorpora los servicios de infraestructura virtuales (VPS) mediante el uso de nube pública, privada, comunitaria o híbrida bajo GNU/Linux (Subashini & Kavitha, 2011).

4. Resultados Arquitectura Microservicios

A partir del análisis de la propuesta aplicada, surge la necesidad de evaluar la eficiencia de las arquitecturas de *software* (3 y 5 capas) propuestas para ambos proyectos de investigación. Por ello, utilizamos las pruebas de rendimiento y estrés para obtener valores cuantitativos para medir el rendimiento general de la aplicación y su interacción con los usuarios finales (Lonetti & Marchetti, 2018). Para tal efecto, fue necesario utilizar un entorno de pruebas controladas a través del uso de VPS (1 vCPUs, 4 GB Memoria, 80 GB Disco, *Docker* con *Ubuntu Server 16.04 LTS*), con el propósito de generar una carga de trabajo que aumenta con el tiempo emulando un número específico de sesiones de usuarios concurrentes (30 usuarios virtuales) acordes a las especificaciones de los VPS y, posteriormente, validar dichos resultados con un criterio de aceptación. Otra tarea prioritaria, es el desarrollo de *scripts* en Apache JMeter, para medir la carga de usuarios virtuales y su comportamiento en el rendimiento de los microservicios, en conjunto con el uso de servidores de *BlazeMeter* (US West - Oregon, USA - Google) (Abdullah et al., 2019; Khan & Amjad, 2016).

Dichas pruebas generan una carga de trabajo que comienza con acceder al inicio de la sesión y desplegar la data médica de los pacientes, correspondiente a la afiliación del usuario final, debido a que ambos casos (Clínica Universitaria y SPIDEP) comparten esta característica. El propósito de estas pruebas es generar varias solicitudes por segundo y que, a su vez, aumenta después de cada minuto de manera progresiva (20 minutos), con una tasa promedio de 15≈20 solicitudes por segundo. La tabla 1 muestra los datos experimentales sobre la implementación de los microservicios para ambos casos.

El reporte contiene varios valores importantes divididos en dos casos (totalidad de los microservicios ejecutados): cantidad de la muestra, tiempo de respuesta promedio, número de solicitudes que se procesan por segundo, percentil 90, porcentajes de solicitudes fallidas, cantidad de usuarios interactuando en las pruebas.

Caso	Muestras	Tiempo Respuesta (ms)	Media. Hits/s	90% línea (ms)	Porciento Errores	Usuario virtual (conurrencia)
CU-UTP	25985	693.2	22.02	867	0%	30
SPIDEP	17868	1007.69	15.14	1239	0%	30

Tabla 1 – Resultados de las pruebas de rendimiento en los casos de clínica universitaria (CU-UTP) y SPIDEP

También se realizaron varias pruebas preliminares para llevar a cabo simulaciones, análisis de los resultados de las ejecuciones y depuraciones de los microservicios que son agregados para crear el sistema principal (*eHealth Cloud*), cuya función es exponer los problemas que pueden surgir en esta etapa de integración de los grupos de microservicios (Abdullah et al., 2019). Sin embargo, los microservicios son aceptables cuando se prueban individualmente en forma aislada. De hecho, podría resultar en un comportamiento incoherente o erróneo cuando se combinan con el fin de construir el conjunto. Por ende, se tomó como referencia el caso CU-UTP para aplicar esta prueba, ya que SPIDEP aún se encuentra en desarrollo Beta. Se procedió a deshabilitar el uso del microservicio de visualización de gráficas (VG-MS) del grupo DCU, pero sin afectar al microservicio enfocado al registro de los reportes clínicos (RRC-MS). El objetivo fue verificar el rendimiento y la integración de VG-MS con una aplicación de tercero (R - JSON Files), para analizar los datos sin depender de la capa de aplicación. Los resultados obtenidos en un análisis preliminar de VG-MS se muestran en la figura 3.

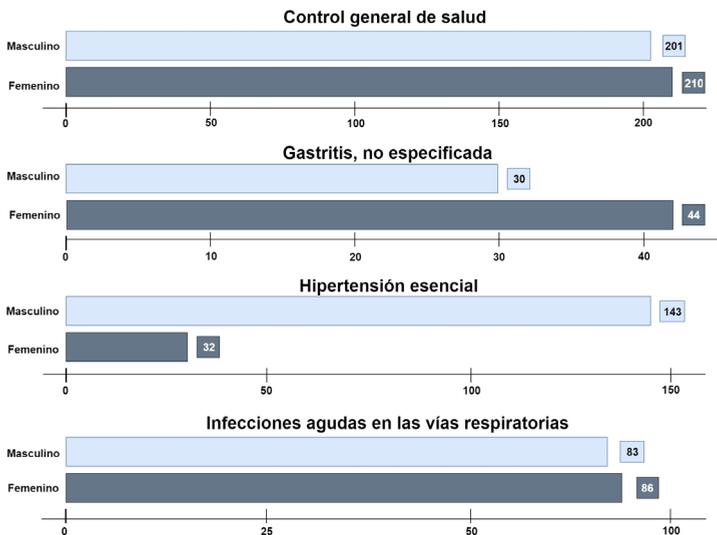


Figura 3 – Ejecución exitosa de las pruebas de integración en la extracción de los datos (2016-2017) a través VG-MS y R. La subgráfica superior representa el principal motivo de las visitas de los pacientes a la clínica universitaria (atención preventiva), y el resto de las subgráficas se basa en las visitas de los pacientes por motivo de enfermedades más frecuentes (tres principales), aplicado a CU-UTP

5. Discusión

Este estudio se propuso con el objetivo de evaluar el rendimiento y el análisis de los microservicios aplicados en arquitecturas de software modificados para tres y cinco capas, dentro del contexto eHealth, por lo que hemos identificado las principales tareas adoptadas para el diseño y el uso de las propuestas para llevar a cabo esta actividad.

Hay varios hallazgos significativos de este estudio: en el primero es importante considerar los criterios de aceptación (CA), como el tiempo de respuesta promedio y percentil 90 (CA-1) y su relación con el porcentaje de solicitudes fallidas (CA-2), con el propósito de validar el rendimiento de los microservicios. Por lo que, mediante el empleo de diversas investigaciones (Khan & Amjad, 2016; Ramakrishnan & Kaur, 2019), se estableció que el límite aceptable para CA-1 es de 5000 ms y el límite aceptable para CA-2 es de 5%. A partir de los límites establecidos para cada salida de las pruebas superior a estos límites, se considera como un rendimiento no aceptable. Teniendo en cuenta lo anterior, a partir de la tabla 1, podemos establecer que las pruebas con 30 usuarios son aceptables, ya que ambos casos cumplen con los dos criterios establecidos. Basándonos en estos resultados, hemos observado que los microservicios implementados sin supervisión (Abdullah et al., 2019) o que no consideran los puntos débiles de esta tecnología (Götz et al., 2018; Ma, Fan, Chuang, Liu, & Lan, 2019), pueden comprometer el rendimiento global del sistema e incluso tener un rendimiento peor que una arquitectura monolítica.

En cuanto al segundo hallazgo, es importante tener en cuenta que nuestra arquitectura de software de tres capas está orientada al desarrollo de CIS intrauniversitario o intraorganizacional que no requieran una gran capacidad de recursos informáticos para su ejecución, mientras que la arquitectura de cinco capas, está orientada a CIS interuniversitario o interorganizacional. Del tercer hallazgo se obtiene una muestra de 829 pacientes (457 hombres y 372 mujeres) atendidos durante los años 2016-2017 de una población de 1452 pacientes (716 mujeres y 736 hombres) del sistema CU-UTP. Cabe señalar que estos datos fueron delimitados a un intervalo de tiempo debido a que los del 2018 fueron recopilados hasta los seis primeros meses, y así evitar un sesgo en el análisis. Lo interesante de estos datos es que nos permiten estudiar distintas categorías (género, edad, región o estamentos) en diferentes periodos (diario, mensual o anual). Como seguimiento de este estudio, en conjunto con los profesionales de la salud, se determinó un punto de referencia para el análisis de las afecciones que más aquejan a los pacientes. Como se observa en la figura 5, existe un alto índice de asistencia de los pacientes para rutina de control general en vez de un tratamiento de alguna afección (gastritis, hipertensión esencial u otra). En general, estos resultados indican que los pacientes tienen la tendencia a practicar la medicina preventiva, la cual se ha mantenido en los últimos años; sin embargo, una dificultad a la que nos hemos enfrentado en este análisis es que no se tomaron en cuenta otros factores externos o antecedentes familiares para determinar las causas de las afecciones.

6. Conclusiones y trabajos futuros

Con el desarrollo del presente estudio se logró completar el objetivo propuesto: obtener dos arquitecturas de software en capas estables y modificables, según las necesidades del entorno de las organizaciones, centradas al escalado ágil de los microservicios que se utilizan en gran medida al replicarlos en varios contenedores sin infrautilizar los recursos

informáticos. De igual manera, esta propuesta da cabida a futuras investigaciones para el diseño de sus propias capas o variantes de esta arquitectura, con el propósito de lograr una mayor eficiencia en la organización y atención de los pacientes. Como trabajos futuros, se tiene contemplada la posibilidad de optimizar aún más el rendimiento de CU-UTP y SPIDEP al implementar los microservicios en otros paradigmas de programación y realizar comparaciones o de extender la arquitectura diseñada en otras capas.

Agradecimiento

Este trabajo forma parte del proyecto AC16 / 00061 “Diseño e implementación de un sistema inteligente de bajo costo para pre-diagnóstico y teleasistencia de enfermedades infecciosas en personas mayores (SPIDEP)” financiado con recursos del Instituto de Salud Carlos III (ISCIII), del Ministerio de Ciencia, Innovación y Universidades de España, de la Fundación para la Investigación Biomédica del Hospital Universitario Príncipe de Asturias (FUHUPA), la Secretaria Nacional de Ciencia, Tecnología e Innovación de Panamá (SENACYT), a través del Sistema Nacional de Investigación (SNI), y dentro de la 2ª Convocatoria Conjunta de Investigación e Innovación ERANet-LAC del Séptimo Programa Marco de Investigación y Desarrollo de Tecnología (FP7) de la Unión Europea bajo el contrato ELAC2015 /T09-0819 SPIDEP.

Contribución de los Autores

Conceptualización, J.G., J.C. y J.S.; metodología, A.G., J.G., J.C. y J.S.; software, H.C. y M.V.; validación H.C. F.N. y M.V.; análisis formal, A.G., H.C., J.G. y M.V.; investigación, A.G., H.C., J.G., J.C., M.V., M.P., G.S. y J.S.; recursos, J.G., M.V., M.P., G.S. y J.S.; datos clínicos, M.P. y G.S.; escritura—original M.V. y H.C.; escritura—revisión y edición, A.G., H.C., J.G., J.C., M.V. y J.P.; visualización, A.G., y H.C.; supervisión, J.G., M.V. y J.S.; administración de proyectos, J.G.; autor de correspondencia MV, Administración de fondos, J.G. y J.S.

Referencias

- Abdullah, M., Iqbal, W., & Erradi, A. (2019). Unsupervised learning approach for web application auto-decomposition into microservices. *Journal of Systems and Software*, 151, 243–257. <https://doi.org/10.1016/J.JSS.2019.02.031>
- Alharbi, F., Atkins, A., & Stanier, C. (2017). Cloud computing adoption in healthcare organisations: a qualitative study in Saudi Arabia. *Transactions on Large Scale Data and Knowledge-centered Systems, XXXV. Lecture Notes in Computer Science*, 10680, 96-131. Berlín: Springer.
- Cooper, A., & Reimann, R. (2004). *About face 2.0: the essentials of interaction design* (2nd ed). Nueva York, NY: Wiley & Sons.
- Catley, C., & Frize, M. (2003). Design of a health care architecture for medical data interoperability and application integration, *Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society, USA* (1952–1953). <https://doi.org/10.1109/iembs.2002.1053111>

- Cho, V., & Chan, A. (2015). An integrative framework of comparing SaaS adoption for core and non-core business operations: An empirical study on Hong Kong industries. *Information Systems Frontiers*, 17(3), 629–644. <https://doi.org/10.1007/s10796-013-9450-9>
- Ciavotta, M., Alge, M., Menato, S., Rovere, D., & Pedrazzoli, P. (2017). A microservice-based middleware for the digital factory. *Procedia Manufacturing*, 11, 931–938. <https://doi.org/10.1016/J.PROMFG.2017.07.197>
- Di Francesco, P., Lago, P., & Malavolta, I. (2019). Architecting with microservices: a systematic mapping study. *Journal of Systems and Software*, 150, 77–97. <https://doi.org/10.1016/J.JSS.2019.01.001>
- Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Communications and Mobile Computing*, 13(18), 1587–1611. <https://doi.org/10.1002/wcm.1203>
- El Kafhali, S., & Salah, K. (2018). Performance analysis of multi-core VMs hosting cloud SaaS applications. *Computer Standards & Interfaces*, 55, 126–135. <https://doi.org/10.1016/J.CSI.2017.07.001>
- Elsayed, M., & Zulkernine, M. (2019). Offering security diagnosis as a service for cloud SaaS applications. *Journal of Information Security and Applications*, 44, 32–48. <https://doi.org/10.1016/J.JISA.2018.11.006>
- Esposito, C., Castiglione, A., Tudorica, C., & Pop, F. (2017). Security and privacy for cloud-based data management in the health network service chain: a microservice approach. *IEEE Communications Magazine*, 55(9), 102–108. <https://doi.org/10.1109/MCOM.2017.1700089>
- Fan, H., Hussain, F. K., Younas, M., & Hussain, O. K. (2015). An integrated personalization framework for SaaS-based cloud services. *Future Generation Computer Systems*, 53, 157–173. <https://doi.org/10.1016/J.FUTURE.2015.05.011>
- Fiedler, B. A. (2017). Challenges of new technology: securing medical devices and their software for HIPPA compliance. *Managing Medical Devices Within a Regulatory Framework*, 315–329. <https://doi.org/10.1016/B978-0-12-804179-6.00018-6>
- Furda, A., Fidge, C., Barros, A., & Zimmermann, O. (2017). Reengineering data-centric information systems for the cloud – a method and architectural patterns promoting multitenancy. *Software Architecture for Big Data and the Cloud*, 227–251. <https://doi.org/10.1016/B978-0-12-805467-3.00013-2>
- Gholap, J., Janeja, V. P., & Yesha, Y. (2015). Unified framework for clinical data analytics (U-CDA). *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015*, 2939–2941. <https://doi.org/10.1109/BigData.2015.7364129>
- Götz, B., Schel, D., Bauer, D., Henkel, C., Einberger, P., & Bauernhansl, T. (2018). Challenges of production microservices. *Procedia CIRP*, 67, 167–172. <https://doi.org/10.1016/J.PROCIR.2017.12.194>

- Handayani, P. W., Hidayanto, A. N., Pinem, A. A., Hapsari, I. C., Sandhyaduhita, P. I., & Budi, I. (2017). Acceptance model of a hospital information system. *International Journal of Medical Informatics*, 99, 11–28. <https://doi.org/10.1016/J.IJMEDINF.2016.12.004>
- Jita, H., & Pieterse, V. (2019). A Framework to apply the internet of things for medical care in a home environment, 45–54. <https://doi.org/10.1145/3291064.3291065>
- Khan, R., & Amjad, M. (2016). Performance testing (load) of web applications based on test case management. *Perspectives in Science*, 8, 355–357. <https://doi.org/10.1016/J.PISC.2016.04.073>
- Kiss, T., Kacsuk, P., Kovacs, J., Rakoczi, B., Hajnal, A., Farkas, A., ... Terstyanszky, G. (2019). MiCADO—Microservice-based cloud application-level dynamic orchestrator. *Future Generation Computer Systems*, 94, 937–946. <https://doi.org/10.1016/J.FUTURE.2017.09.050>
- Krämer, M., Frese, S., & Kuijper, A. (2019). Implementing secure applications in smart city clouds using microservices. *Future Generation Computer Systems*, 99, 308–320. <https://doi.org/10.1016/J.FUTURE.2019.04.042>
- Kumar, N. M., & Mallick, P. K. (2018). The Internet of Things: Insights into the building blocks, component interactions, and architecture layers. *Procedia Computer Science*, 132, 109–117. <https://doi.org/10.1016/J.PROCS.2018.05.170>
- Leung, L., & Chen, C. (2019). E-health/m-health adoption and lifestyle improvements: Exploring the roles of technology readiness, the expectation-confirmation model, and health-related information activities. *Telecommunications Policy*, 43(6), 563–575. <https://doi.org/10.1016/J.TELPOL.2019.01.005>
- Li, S., Zhang, H., Jia, Z., Li, Z., Zhang, C., Li, J., & Gao, Q. (2019). A dataflow-driven approach to identifying microservices from monolithic applications. *Journal of Systems and Software*. <https://doi.org/10.1016/J.JSS.2019.07.008>
- Lonetti, F., & Marchetti, E. (2018). Emerging software testing technologies. *Advances in Computers*, 108, 91–143. <https://doi.org/10.1016/BS.ADCOM.2017.11.003>
- Ma, S.-P., Fan, C.-Y., Chuang, Y., Liu, I.-H., & Lan, C.-W. (2019). Graph-based and scenario-driven microservice analysis, retrieval, and testing. *Future Generation Computer Systems*, 100, 724–735. <https://doi.org/10.1016/J.FUTURE.2019.05.048>
- Mousa, A., Bentahar, J., & Alam, O. (2019). Context-aware composite SaaS using feature model. *Future Generation Computer Systems*. <https://doi.org/10.1016/J.FUTURE.2019.04.032>
- Newman, S. (2015). *Building microservices: designing fine-grained systems*. O'Reilly Media, Inc.
- Oliveira, T., Martins, R., Sarker, S., Thomas, M., & Popovič, A. (2019). Understanding SaaS adoption: The moderating impact of the environment context. *International Journal of Information Management*, 49, 1–12. <https://doi.org/10.1016/J.IJINFOMGT.2019.02.009>

- Organización Mundial de la Salud. (2019). Acerca de los sistemas de salud. Retrieved from <https://www.who.int/healthsystems/about/es/>
- Organización Panamericana de la Salud. (2011). Estrategia y plan de acción sobre eSalud (2012-2017). *51.º Consejo Directivo de La OPS, 63.ª Sesión Del Comité Regional de La OMS Para Las Américas; Del 26 Al 30 de Septiembre Del 2011 (Documento CD51/13), 28(2005), 25.*
- Raković, P., & Lutovac, B. (2015). A cloud computing architecture with wireless body area network for professional athletes health monitoring in sports organizations- Case study of Montenegro. *Proceedings - 2015 4th Mediterranean Conference on Embedded Computing, MECO 2015 - Including ECyPS 2015, BioEMIS 2015, BioICT 2015, MECO-Student Challenge 2015, 387–390.* <https://doi.org/10.1109/MECO.2015.7181950>
- Ramakrishnan, R., & Kaur, A. (2019). Little's law based validation framework for load testing. *Information and Software Technology, 108, 88–98.* <https://doi.org/10.1016/J.INFSOF.2018.11.007>
- Saadeh, M., Sleit, A., Sabri, K. E., & Almobaideen, W. (2018). Hierarchical architecture and protocol for mobile object authentication in the context of IoT smart cities. *Journal of Network and Computer Applications, 121, 1–19.* <https://doi.org/10.1016/J.JNCA.2018.07.009>
- Sharma, A., Kumar, M., & Agarwal, S. (2015). A complete survey on software architectural styles and patterns. *Procedia Computer Science, 70, 16–28.* <https://doi.org/10.1016/j.procs.2015.10.019>
- Stavrinides, G. L., & Karatza, H. D. (2019). Performance evaluation of a SaaS cloud under different levels of workload computational demand variability and tardiness bounds. *Simulation Modelling Practice and Theory, 91, 1–12.* <https://doi.org/10.1016/J.SIMPAT.2018.11.006>
- Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications, 34(1), 1–11.* <https://doi.org/10.1016/J.JNCA.2010.07.006>
- Vilaplana, J., Solsona, F. F., Abella, Filgueira, R., & Rius, J. (2013). The cloud paradigm applied to e-Health. *BMC Medical Informatics and Decision Making, 13(1), 1–10.* <https://doi.org/10.1186/1472-6947-13-35>
- Yogesh, S., Pushkara, R., Shilpa, C., Malati, H., & Rashmi, B. (2018). Towards a data-driven IoT software architecture for smart city utilities. *Software: Practice and Experience, 48(7), 1390–1416.* <https://doi.org/10.1002/spe.2580>