Programa de Doctorado en Ingeniería de la Información y del Conocimiento (D442)

DISEÑO E IMPLEMENTACIÓN DE UNA ARQUITECTURA DE CAPAS ENFOCADA A MICROSERVICIOS EN EL CONTEXTO EHEALTH

Tesis Doctoral presentada por

HURIVIADES CALDERÓN GÓMEZ

Directores: Dr. MIGUEL VARGAS LOMBARDO y Dr. JOSÉ MANUEL GÓMEZ PULIDO

Alcalá de Henares, Madrid, 2023

Vt sementem feceris, ita metes

Cicerón, De oratore II 261

Agradecimientos

Primeramente, me gustaría agradecer a mis directores de tesis al Dr. José Manuel Pulido quien no escatimo esfuerzo para guiarme y apoyarme a lo largo de las tesis. También, agradecerle al Dr. Miguel Vargas por la confianza depositada y la paciencia de guiarme con esta tesis doctoral.

Igualmente, agradezco al Dr. Oscar Gutiérrez Blanco por su apoyo incondicional, dedicación y disponibilidad que tuvo en todo mi trayecto como doctorando. Además, agradecerles a las personas del grupo de Computación en Salud y Sistemas Inteligentes de la Universidad de Alcalá (Dr. José Luis Castillo Sequera, Dr. Alberto Garcés Jiménez y al Dr. Jesús Escobar Bentué) y al departamento de Ciencias de la Computación por su apoyo y ayuda durante estos años.

También, agradezco al Ing. Luis Mendoza Pittí por su apoyo y amistad incondicional en este curso de mi vida.

Agradezco a mis padres, mi hermano, mis amigos y mis compañeros del trabajo que me apoyaron, creyeron y confiaron en mí en todo momento para concluir con éxito este doctorado.

Resumen

Esta memoria de tesis doctoral tiene el objetivo primordial de proponer una arquitectura multicapa dirigida a soportar la interoperabilidad entre las aplicaciones de salud tradicionales involucradas en las instituciones médicas (p. ej. ficha clínica digital), comúnmente desarrolladas bajo el estilo arquitectónico Service-Oriented Architecture "SOA", y en conjunto con aplicaciones modernas bajo el estilo arquitectónico Microservice Architecture "MSA" adaptados a variantes Basado en aprendizaje automático mediante el uso de metadatos estandarizados proveniente de diversas fuentes de datos dinámicos y heterogéneos. Sin embargo, estos rápidos cambios han tenido abrumadores impactos de cómo se relaciona la salud con el mundo digital causado por la tendencia de la digitalización e impulsado por políticas de un gobierno electrónico.

A diferencia de otras propuestas de investigación relacionadas a la arquitectura de microservicios, esta propuesta se centra en la replicabilidad, escalabilidad e interoperabilidad de los servicios especializados que conforman a una propia característica de software basada en la **versatilidad de alto rendimiento**. Para demostrar la viabilidad del ecosistema de salud digital fue necesario adaptar esta propuesta a un caso de uso real, específicamente al proyecto SPIDEP (Diseño e implementación de un sistema inteligente de bajo coste para el prediagnóstico y la teleasistencia de enfermedades infecciosas en personas de edad avanzada) perteneciente a la convocatoria internacional ERANET LAC 2015-FP7, cuya finalidad del proyecto fue construir un sistema inteligente basado en las tecnologías de la información y las comunicaciones para apoyar el diagnóstico temprano de enfermedades infecciosas respiratorias y urinarias en personas mayores a través de la recolección y seguimiento remoto de los pacientes en residencias, lo que permitió la detección de anomalías en los signos vitales.

Teniendo en cuenta lo anterior, mediante esta memoria se han realizado cincos contribuciones principales: (i) demostrar las implicaciones y desafíos que conlleva implementar esta arquitectura multicapa orientada a microservicios en un ecosistema de salud digital; (ii) proveer los pasos necesarios para el diseño, implementación y despliegue de esta propuesta adaptado a un caso de uso exitoso mediante una plataforma SPIDEP; (iii) señalar que esta propuesta es respaldada por una gran base de datos que integra fuentes de información diversas y heterogéneas obtenida en la duración de SPIDEP; (iv) definir las característica de softwares centrada en solventar las necesidades de la televigilancia médica enfocado a arquitectura multicapa; (v) que la implementación de SOA y MSA depende de la naturaleza y las necesidades de las organizaciones médicas (p. ej. rendimiento, interoperabilidad u otros). Sin embargo, los patrones arquitectónicos SOA y MSA pueden considerarse aliados complementarios para una arquitectura interempresarial o intergeneracional que

confiere un conjunto de diferentes servicios, en lugar de ser competidores. Cabe destacar, que esta propuesta se creó con miras de ser adaptado a otras áreas eHealth (p. ej. diálisis, diabetes, cáncer de colon u otros).

Por otro lado, esta memoria es presentada bajo la modalidad de compendio basándose en tres contribuciones científicas publicadas en prestigiosas revistas que corresponden a sendos trabajos de investigación que reflejan mediante un sólido hilo conductor la aportación original de la tesis para proporcionar servicios inteligentes avanzados en el ámbito de un ecosistema de salud digital desarrollando a tal efecto una propuesta de arquitectura de varias capas que soporta novedoso y avanzados microservicios especializados, cuyas publicaciones son: (i) *Medical Prognosis of Infectious Diseases in Nursing Homes by Applying Machine Learning on Clinical Data Collected in Cloud Microservices*, este primer artículo se centra en la propuesta inicial de una arquitectura de microservicios flexible que proporcione el acceso y la funcionalidad al sistema orientado a un ecosistema de salud digital en el que se pueda realizar el tratamiento de las enfermedades infecciosas en las personas de la tercera edad, ya que estos pacientes tienden a llegar a las consultas médicas con síntomas avanzados mediante el uso de algoritmos de aprendizaje automático basado en clasificadores para reconocer patrones a través de biosensores no invasivos; (ii) *Telemonitoring System for Infectious Disease Prediction in Elderly People Based on a Novel Microservice Architecture*, este segundo artículo se describe el diseño, desarrollo e implementación de nueves servicios enfocado a una arquitectura en microservicios, que permite la detección y el diagnóstico clínico asistido dentro del campo de las enfermedades infecciosas de pacientes de tercera edad, basándose en el uso de la televigilancia. A diferencia con el primer artículo, en este nos enfocamos más en el aspecto de Ingeniería de Software, ya que proponemos un flujo de trabajo más maduro para el despliegue continuo y automatización de los microservicios desarrollados; (iii) *Evaluating Service-Oriented and Microservice Architecture Patterns to Deploy eHealth Applications in Cloud Computing Environment*, este tercer artículo propone un nuevo marco de trabajo para la concepción de una plataforma eHealth centrada en los entornos de computación en la nube, ya que los enfoques actuales y emergentes con respecto al desarrollo de sistemas de recomendación basados en la telemonitorización y el acceso a la historia digital clínica para las diferentes áreas de la salud. A diferencia de los dos artículos anteriores, este evaluó y contrastó el rendimiento de los diferentes patrones arquitectónicos más utilizados para la creación de aplicaciones de salud tanto en su variante SOA y MSA, tomando como referencia los valores cuantitativos obtenidos de las diversas pruebas de rendimiento y a su capacidad de adaptarse a las características de softwares requeridas en SPIDEP. Como resultado, se determinó que MSA presenta un mejor desempeño en cuanto al atributo de calidad de rendimiento (~54.21%), de la misma manera al procesar múltiples solicitudes de diversos servicios el tiempo de respuesta fue menor en comparación con SOA (~7.34%), pero el consumo de ancho de banda en MSA fue más significativo que SOA (~73.80%).

Abstract

This doctoral thesis aims to primarily propose a multi-layer architecture aimed at supporting interoperability between traditional health applications involved in medical institutions (e.g., digital medical record), commonly developed under the Service-Oriented Architecture (SOA) architectural style, and in conjunction with modern applications under the Microservice Architecture (MSA) adapted to variations based on machine learning through the use of standardized metadata from various sources of dynamic and heterogeneous data. However, these rapid changes have had overwhelming impacts on how health relates to the digital world, caused by the digitization trend and driven by e-government policies.

Unlike other research proposals related to microservices architecture, this proposal focuses on the replicability, scalability, and interoperability of specialized services that make up a high-performance versatility-based software feature. To demonstrate the feasibility of the digital health ecosystem, it was necessary to adapt this proposal to a real-world use case, specifically the SPIDEP project (Design and implementation of a low-cost intelligent system for the pre-diagnosis and teleassistance of infectious diseases in elderly people) belonging to the international ERANET LAC 2015-FP7 call, whose goal was to build an intelligent system based on information and communication technologies to support early diagnosis of respiratory and urinary infectious diseases in older people through the remote collection and monitoring of patients in nursing homes, which allowed for the detection of anomalies in vital signs.

On this basis, this thesis has made five main contributions: (i) demonstrate the implications and challenges involved in implementing this microservices-oriented multi-layer architecture in a digital health ecosystem; (ii) provide the necessary steps for the design, implementation, and deployment of this proposal adapted to a successful use case through the SPIDEP platform; (iii) indicate that this proposal is supported by a large database that integrates various and heterogeneous sources of information obtained during the SPIDEP project; (iv) define the software features focused on solving the needs of medical telemonitoring with a multi-layer architecture; (v) and that the implementation of SOA and MSA depends on the nature and needs of medical organizations (e.g. performance, interoperability, or others) and that the implementation of SOA and MSA depends on the nature and needs of medical organizations (e. v. However, the SOA and MSA architectural patterns can be considered complementary allies for an inter-enterprise or inter-generational architecture that confers a set of different services rather than being competitors. It's worth mentioning that this proposal was created with the intention of being adapted to other eHealth areas (e.g., dialysis, diabetes, colon cancer, or others).

On the other hand, this thesis is presented in the form of a compendium based on three scientific contributions published in prestigious journals, which correspond to respective research works that reflect, through a solid thread, the original contribution of the thesis to provide advanced intelligent services in the field of a digital health ecosystem, by developing in this way a proposal of multi-layer architecture that supports advanced specialized microservices, whose publications are: (i) Medical Prognosis of Infectious Diseases in Nursing Homes Using Machine Learning on Clinical Data Collected in Cloud Microservices, the first article focuses on the initial proposal of a flexible microservices architecture that provides access and functionality to the system oriented to a digital health ecosystem in which it can be made to treat infectious diseases in elderly people because these patients tend to arrive at medical appointments with advanced symptoms. (ii) Telemonitoring System for Infectious Disease Prediction in Elderly People Based on a novel microservice architecture, this second article describes the design, development, and implementation of new services focused on a microservices architecture that allows the detection and clinically assisted diagnosis of infectious diseases in elderly patients based on the use of telemonitoring. Unlike the first article, in this one we focus more on the aspect of software engineering, as we propose a more mature workflow for the continuous deployment and automation of the developed microservices. (iii) Evaluating Service-Oriented and Microservice Architecture Patterns to Deploy eHealth Applications in a Cloud Computing Environment. This third article proposes a new framework for the conception of an eHealth platform focused on cloud computing environments, given current and emerging approaches regarding the development of telemonitoring-based recommendation systems and access to digital clinical history for different health areas. Unlike the previous two articles, this one evaluates and compares the performance of the most used architectural patterns for developing health applications, both in their SOA and MSA variants, using as a reference the quantitative values obtained from the various performance tests and its ability to adapt to the software characteristics required by SPIDEP. As a result, it was determined that MSA presents better performance in terms of the performance quality attribute (54.21%), in the same way that when processing multiple requests from different services, the response time was lower in comparison with SOA (7.34%), but the bandwidth consumption in MSA was more significant than SOA (73.80%)

Índice

1 Introducción 11

1.1 Hipótesis de la investigación 13

1.2 Objetivos de la investigación 14

1.2.1 Objetivos general 14

1.2.2 Objetivos específicos 14

1.2.3 Estructura de la memoria de tesis doctoral 14

2 Compendio de artículos 15

2.1 Artículo 1: "Medical Prognosis of Infectious Diseases in Nursing Homes by Applying Machine Learning on Clinical Data Collected in Cloud Microservices" 16

2.1.1 Resumen del artículo 16

2.1.2 Conclusiones del artículo 17

2.1.3 Contribuciones del artículo a la comunidad científica 17

2.1.4 Identificación del artículo 17

2.1.5 Índices de calidad del artículo 18

2.1.6 Artículo publicado 18

2.2 Artículo 2: "Telemonitoring System for Infectious Disease Prediction in Elderly People Based on a Novel Microservice Architecture" 35

2.2.1 Resumen del artículo 35

2.2.2 Conclusiones del artículo 35

2.2.3 Contribuciones del artículo a la comunidad científica 36

2.2.4 Identificación del artículo 36

2.2.5 Índices de calidad del artículo 36

2.2.6 Artículo publicado 36

2.3 Artículo 3: "Evaluating Service-Oriented and Microservice Architecture Patterns to Deploy eHealth Applications in Cloud Computing Environment" 52

2.3.1 Resumen del artículo 52

2.3.2 Conclusiones del artículo 52

2.3.3 Contribuciones del artículo a la comunidad científica 53

2.3.4 Identificación del artículo 54

2.3.5 Índices de calidad del artículo 54

2.3.6 Artículo publicado 54

3 Otros Méritos 83

3.1 Congresos Internacionales 83

3.1.1 LACAR 2019: Latin American Congress on Automation and Robotics 2019........83

3.1.2 CSEI 2019: I Conference on computer science, electronics, and industrial engineering..83

3.1.3 2019 7th International Engineering, Sciences and Technology Conference (IESTEC) 84

3.2 Colaboración con proyectos de investigación 84

3.2.1 SPIDEP 84

3.2.2 CitizenLab 84

4 Conclusiones y trabajos futuros.. 85

5 Referencias... 87

1 Introducción

En los últimos años, ha habido un interés creciente en la cloud-native computing [1]–[3] es un novedoso concepto para el desarrollo, implementación y operación de aplicaciones que se ejecutan en un entorno de nube bajo la tecnologías de contenedores y orquestadores [4]. Este concepto tiene como propósito que el personal TI tenga más tiempo para esforzarse en la creación de aplicaciones de calidad en vez de resolver problemas tediosos con la infraestructura [5], [6]. Es por ello, que las aplicaciones modernas creada bajo los paradigmas computacionales como Edge, Fog o IoT se desarrollan cada vez más siguiendo este concepto [7]–[12]; sin embargo, para que las nuevas aplicaciones aproveche al máximo la computación distribuida en nube es necesario contar con una arquitectura de software que soporte todo el ecosistema de aplicaciones.

Vinculado a lo anterior, cada vez es mayor el uso masivo de la Big Data lo que contribuyen al crecimiento exponencial de los datos y a su vez ha provocado la adopción del Machine Learning (ML) para analizar patrones y tendencias en los comportamiento de las personas y proveer contenido más personalizado y preciso [5], [8], [13], [14], especialmente en el sector de la salud [10], [15], [16]. No obstante, muchas aplicaciones eHealth utilizan el estilo SOA lo cual dificulta la reutilización de los componentes y en consecuencia no se proporcionan los mecanismos necesarios para la interoperabilidad de los datos entre las diversas plataformas (p. e.j. telemonitoreo, chatbot u otros) [16]–[24], y a su vez provoca una falta de flexibilidad y escalabilidad en la comunicación directa entre los nodos de la red, lo cual es fundamental para enfrentar los desafíos de la producción en masa altamente personalizada [25]. Asimismo, al ejecutar aplicaciones o acceder a servicios web, tanto los usuarios como los proveedores de TI requieren que estas aplicaciones o servicios se ajustan dinámicamente a las fluctuaciones de la demanda y ofrezcan a los usuarios la calidad de servicio requerida (p. e.j. rendimiento, confiabilidad, seguridad, entre otros) ligado a un costo optimizado [26]–[28].

Es por ello, que los microservicios ha recibido atención significativa desde el punto de vista de los investigadores en las diferentes áreas del conocimiento [27], [29], [30]; sin embargo, hay pocas investigaciones enfocadas al área de eHealth o su equivalente al español eSalud. Por lo tanto, la contribución de esta tesis va dirigida al contexto de la salud, específicamente a consolidar los procesos de prestación de servicios de salud, realizando una reingeniería de los procesos clínicos, simplificando y mejorando el apoyo a las tareas de diagnóstico del especialista médico. Adicionalmente, es importante conocer los aspectos que conforman la salud y su relación con las tecnologías digitales (p. ej. inteligencia artificial, telemedicina, automatización, historia clínica electrónica u otros), ya que es importante comprender la dirección y el impacto que estas nuevas tecnologías influyen en la toma de decisiones en la prevención, monitoreo y tratamiento de enfermedades [31], [32].

Teniendo en cuenta lo anterior, es importante contextualizar estas tendencias desde el punto de vista de la Ingeniería de Software y entender cómo vincular la digitalización y la salud mediante un flujograma maduro para el diseño, implementación y despliegue de servicios vinculado a diversas tecnologías, como se muestra en la Figura 1.

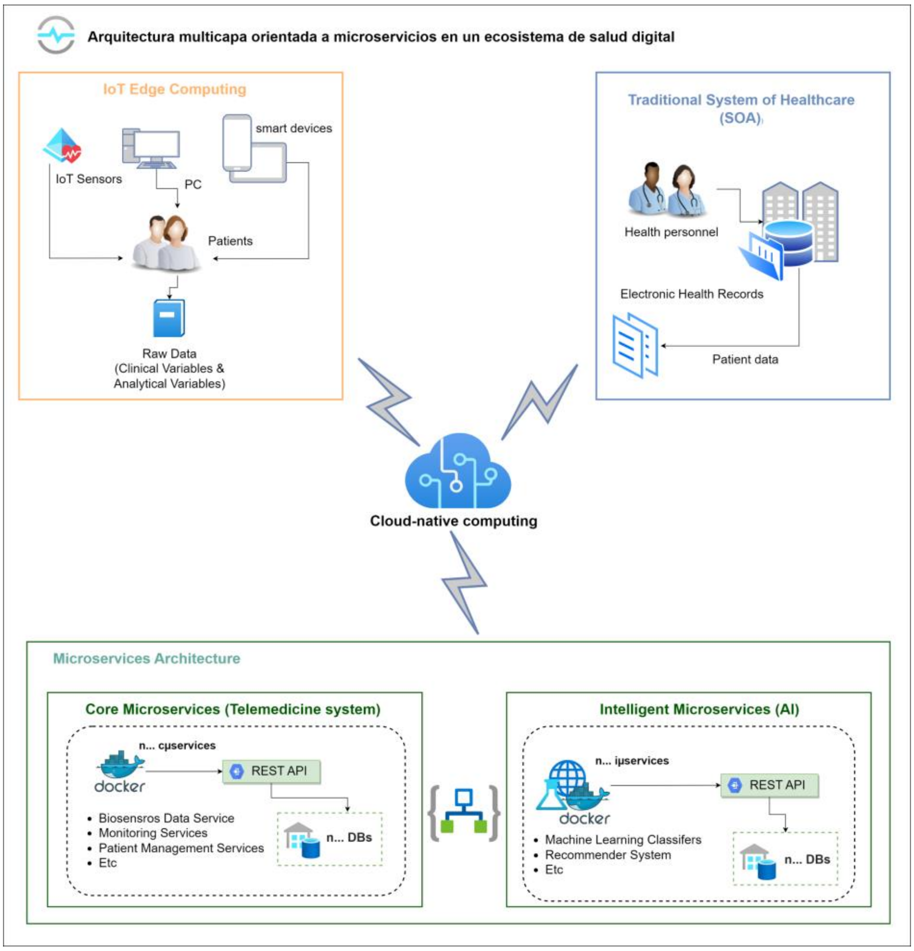

Figura 1. *Esquema general de la arquitectura multicapa orientada a microservicios en un ecosistema de salud digital.*

Dentro de este marco, el emergente ecosistema de salud digital proporcionará movilidad con dispositivos portátiles, wearables o teléfonos inteligentes, servicios ubicuos y recursos adaptables con la nube; mientras que, la investigación en eHealth irá extendiendo el uso de los registros electrónicos, las entradas de órdenes informatizadas, la prescripción electrónica, los sistemas de apoyo a las decisiones clínicas (CDSS), la telemedicina, la gestión del conocimiento sanitario, el triaje, los equipos sanitarios virtuales y los servicios de movilidad médica (mHealth) entre otras aplicaciones [33]–[35]. Sin embargo, las implantaciones tecnológicas podrían entrar en conflicto con los facultativos y el personal de enfermería en algunos casos, ya que conllevan cargas de trabajo adicionales [36], [37].

Teniendo en cuenta lo anterior, fue necesario realizar una investigación exhaustiva sobre las problemáticas que más afectan a las personas, según las necesidades de los proyectos de investigación orientados al sector de la salud pública; por consiguiente, se elaboraron tres artículos previos para definir la problemática principal a tratar [38]–[40]; por consiguiente, se identificó un crecimiento constante de la población envejecida en Europa, por lo que se prevé un crecimiento del 74% de la población mayor de 65 años hasta 2060 [41], [42]. En consecuencia, este grupo de personas tienen una elevada comorbilidad [43], lo que condiciona un consumo muy elevado de recursos tanto de atención primaria como especializada [44], y a su vez una elevada frecuentación de los servicios de urgencias y la necesidad de estancias numerosas y prolongadas en centros especializados. Adicionalmente, esta población son las más frecuentes al ingreso de urgencia por motivos de infecciones, especialmente las respiratorias y urinarias [45]–[49].

Por todo lo anteriormente apuntado he considerado abordar esta gran problemática como objetivo principal para esta tesis doctoral, ya que es necesario la creación de un nuevo marco de trabajo para la concepción de una plataforma eHealth, basándose en la arquitectura de microservicios, que permita la detección y el diagnóstico temprano clínico asistido dentro del campo de las enfermedades infecciosas de pacientes de tercera edad, orientado en el uso de la telemonitorización, cuya meta es de minimizar la severidad de los procesos infecciosos y, colateralmente, la reducción en los recursos necesarios para controlar adecuadamente el problema.

1.1 Hipótesis de la investigación

La hipótesis se formula en base en los siguientes términos:

"Proporcionar servicios de salud y cuidados en personas en riesgo o con difícil acceso a una atención médica preventiva, mediante el uso de nuevas tecnologías que permita lograr una mayor eficiencia en la organización y atención a grupos de población especiales, con el valor añadido de una reducción de costes."

Partiendo de los supuestos anteriores, se tiene tres posibles resultados:

- Hipótesis de la Investigación: el diseño arquitectónico enfocado a los microservicios mejora la eficiencia y el impacto de los servicios de eHealth, basándose en los criterios de aceptación [50], [51].
- Hipótesis Nula: el diseño arquitectónico enfocado a los microservicios no mejora la eficiencia y el impacto de los servicios de eHealth, basándose en los criterios de aceptación [50], [51].

- Hipótesis Alternativa: el diseño arquitectónico enfocado a los microservicios empeora o degrada la eficiencia y el impacto de los servicios de eHealth, basándose en los criterios de aceptación [50], [51].

1.2 Objetivos de la investigación

1.2.1 Objetivos general

Proponer un diseño arquitectónico enfocado a los microservicios, aplicados en arquitecturas de software basadas en capas dentro del contexto eHealth, capaz de mejorar los servicios de salud desde el punto de vista de la gestión, la tecnología, la seguridad y la legalidad.

1.2.2 Objetivos específicos

I. Examinar el uso de los microservicios en el contexto de eHealth.

II. Identificar las principales tareas adoptadas en la creación de un nuevo marco de trabajo para la concepción de plataforma eHealth, basándose en microservicios.

III. Describir las etapas para el desarrollo de la arquitectura de software en capas, orientados a los microservicios.

IV. Evaluar y analizar el rendimiento de los microservicios aplicados en la arquitectura de software basadas en capas, según los criterios de aceptación.

V. Determinar los puntos fuertes y débiles de este enfoque de microservicios.

VI. Demostrar que las nuevas tecnologías emergentes de servicios enfocados a las estrategias de eHealth, contribuyen a una mejor prestación en la gestión y la atención médica.

VII. Determinar los puntos fuertes y débiles de este enfoque de microservicios.

VIII. Demostrar que las nuevas tecnologías emergentes de servicios enfocados a las estrategias de eHealth, contribuyen a una mejor prestación en la gestión y la atención médica.

1.2.3 Estructura de la memoria de tesis doctoral

La presente tesis doctoral se presentará bajo la modalidad de **Tesis por compendio**, debidamente regulada en la normativa por la Universidad de Alcalá establece que [52], [53], "*Si la Comisión Académica del Programa lo autoriza, la Tesis Doctoral podrá realizarse mediante el compendio de artículos del doctorando en publicaciones de reconocido prestigio. El número mínimo de artículos será de tres. La Tesis deberá incluir, además de los artículos, un resumen amplio que dé coherencia al conjunto de la investigación, en el que se muestre la línea argumental de la misma, así como un capítulo de conclusiones. Se entenderá por publicaciones de reconocido prestigio las utilizadas para la obtención de complementos de investigación (sexenios) en el ámbito en el que se desarrolle la investigación.*"

Atendiendo a estas consideraciones, se ha seleccionado tres artículos alineado a la normativa de UAH y que se encuentra indexado al Journal Citation Report (JCR) de Clarivate Analytics, donde el tesista ha realizado las aportaciones científicas alineadas con las hipótesis y objetivos antes señaladas y que serán detallados en la descripción de las publicaciones seleccionadas.

- Garcés-Jiménez, A.; Calderón-Gómez, H.; Gómez-Pulido, J.M.; Gómez-Pulido, J.A.; Vargas-Lombardo, M.; Castillo-Sequera, J.L.; Aguirre, M.P.; Sanz-Moreno, J.; Polo-Luque, M.-L.; Rodríguez-Puyol, D. Medical Prognosis of Infectious Diseases in Nursing Homes by Applying Machine Learning on Clinical Data Collected in Cloud Microservices. Int. J. Environ. Res. Public Health 2021, 18, 13278. https://doi.org/10.3390/ijerph182413278.
- H. Calderón-Gómez et al., "Telemonitoring System for Infectious Disease Prediction in Elderly People Based on a Novel Microservice Architecture," in IEEE Access, vol. 8, pp. 118340-118354, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3005638.
- Calderón-Gómez, H.; Mendoza-Pittí, L.; Vargas-Lombardo, M.; Gómez-Pulido, J.M.; Rodríguez-Puyol, D.; Sención, G.; Polo-Luque, M.-L. Evaluating Service-Oriented and Microservice Architecture Patterns to Deploy eHealth Applications in Cloud Computing Environment. Appl. Sci. 2021, 11, 4350. https://doi.org/10.3390/app11104350.

Adicionalmente de los tres artículos seleccionados para este compendio, se elaboraron tres artículos previos de carácter exploratorio para validar la hipótesis planteada mediante resultados intermedios. A pesar de no haber sido seleccionado para formar parte del compendio, a causa de que estos artículos solo se encuentran indexado en SCimago Journal Record (SJR) de SCimago, fueron importantes para la validación y aceptación de la investigación ante la comunidad científica [38]–[40].

Esta tesis está organizada de la siguiente manera: la sección 2 presenta los artículos seleccionados que conforman el compendio, incluyendo los resúmenes y los detalles de la publicación. La sección 3 menciona otros méritos conseguidos hasta la fecha relacionados con la tesis. Finalmente, la sección 4 proporciona las conclusiones de esta tesis, analiza su potencial y sugiere futuras actividades de investigación.

2 Compendio de artículos

Para garantizar que los artículos publicados han cumplido con los objetivos planteados en este compendio, se ha elaborado una tabla que cuenta con estos objetivos y se destaca con una “X” en el recuadro según el objetivo completado de acuerdo con las publicaciones.

Artículos	**Obj. I**	**Obj. II**	**Obj. III**	**Obj. IV**	**Obj. V**	**Obj. VI**	**Obj. VII**	**Obj. VIII**
Medical Prognosis of Infectious Diseases in Nursing Homes by Applying Machine Learning on Clinical Data Collected in Cloud Microservices	X	X				X		
Telemonitoring System for Infectious Disease Prediction in Elderly People Based on a Novel Microservice Architecture		X	X	X	X	X		
Evaluating Service-Oriented and Microservice Architecture Patterns to Deploy eHealth Applications in Cloud Computing Environment			X	X	X	X	X	X

2.1 Artículo 1: "Medical Prognosis of Infectious Diseases in Nursing Homes by Applying Machine Learning on Clinical Data Collected in Cloud Microservices"

2.1.1 Resumen del artículo

Antecedentes: el tratamiento de las enfermedades infecciosas en personas de edad avanzada es difícil; es frecuente la derivación de pacientes a los servicios de urgencias, ya que los ancianos suelen llegar a las consultas con síntomas avanzados y graves. Objetivo: se planteó la hipótesis de que anticiparse unos días al diagnóstico de una enfermedad infecciosa podría mejorar significativamente el bienestar del paciente y reducir la carga de los servicios del sistema sanitario de urgencias. Métodos: se tomaron diariamente las constantes vitales de los residentes y se transfirieron a una base de datos en la nube. Se utilizaron clasificadores para reconocer patrones en el proceso de dominio espacial de los datos recogidos. Los médicos comunicaban sus diagnósticos cuando se presentaba alguna enfermedad. Una arquitectura de microservicios flexible proporcionó acceso y funcionalidad al sistema. Resultados: combinar dos dominios diferentes, salud y tecnología, no es fácil, pero los resultados son alentadores. Los clasificadores dieron buenos resultados; el sistema ha sido bien aceptado por el personal médico y está demostrando ser rentable y una buena solución para dar servicio a zonas desfavorecidas. En este contexto, esta investigación constató la importancia de determinadas variables clínicas en la identificación de enfermedades infecciosas. Conclusiones: este trabajo explora cómo aplicar las comunicaciones móviles, los servicios en la nube y la tecnología de aprendizaje automático para proporcionar herramientas eficientes al personal médico de las residencias de ancianos. La arquitectura escalable puede extenderse a aplicaciones de big data que pueden extraer patrones de conocimiento valiosos para la investigación médica.

2.1.2 Conclusiones del artículo

Esta investigación propone un sistema de monitorización de eHealth cómodo, flexible, accesible y económico para pacientes institucionalizados en residencias de ancianos, y analiza la predictibilidad de enfermedades infecciosas a partir de las constantes vitales recogidas en el estudio piloto. La arquitectura implementada basada en microservicios es de un bajo coste y escalable. Las pruebas de rendimiento indicaron los límites del sistema en cuanto a su saturación. Además, se probó la funcionalidad de los microservicios para llevar a cabo el servicio óptimo. Adicionalmente, este estudio preliminar arrojó resultados valiosos para la validación del proyecto SPIDEP y fue un buen punto de partida para las posteriores investigaciones relacionadas a los microservicios dentro en un entorno eHealth.

2.1.3 Contribuciones del artículo a la comunidad científica

El primer artículo se propuso, implementó y aplicó una arquitectura de microservicios flexible que proporcionó el acceso y la funcionalidad al sistema orientado a un ecosistema de salud digital en el que se pueda realizar el tratamiento de las enfermedades infecciosas en las personas de la tercera edad, ya que estos pacientes tienden a llegar a las consultas médicas con síntomas avanzados mediante el uso de algoritmos de aprendizaje automático basado en clasificadores para reconocer patrones a través de biosensores no invasivos. Cabe destacar, que el tesista diseño la arquitectura de software de microservicios microservicio bajo una versión temprana o preliminar [54] para el diagnóstico anticipado de enfermedades infecciosas y validó la eficiencia de este a través de pruebas de carga de trabajo normalizados; mientras que, los coautores desarrollaron y validaron las comunicaciones móviles y la tecnología de aprendizaje automático mediante el uso de la big data.

2.1.4 Identificación del artículo

- Título: “Medical Prognosis of Infectious Diseases in Nursing Homes by Applying Machine Learning on Clinical Data Collected in Cloud Microservices”
- DOI: https://doi.org/10.3390/ijerph182413278
- Autores: Alberto Garcés-Jiménez; Huriviades Calderón-Gómez; José M. Gómez-Pulido; Juan A. Gómez-Pulido; Miguel Vargas-Lombardo; José L.

Castillo-Sequera; Miguel Pablo Aguirre; José Sanz-Moreno; María-Luz Polo-Luque; Diego Rodríguez-Puyol
- Acceso del artículo: Open Access (OA)
- Fecha de publicación: 16 de diciembre de 2021
- Idioma: Inglés

2.1.5 Índices de calidad del artículo

- Revista: International Journal of Environmental Research and Public Health (IJERPH)
- ISSN: 1660-4601
- Editor: MDPI
- Volumen y número: 8-24
- Índice de impacto y ranking JCR (2021): 4.614
    - Environmental sciences – Q2
    - Public, environmental & occupational health – Q2
- Índice de impacto y ranking SJR (2021): 0.814
    - Health, toxicology, and mutagenesis – Q1
    - Public health, environmental and occupational health – Q2

2.1.6 Artículo publicado

International Journal of *Environmental Research and Public Health*

MDPI

Article

Medical Prognosis of Infectious Diseases in Nursing Homes by Applying Machine Learning on Clinical Data Collected in Cloud Microservices

Alberto Garcés-Jiménez [1,2], Huriviades Calderón-Gómez [3,4], José M. Gómez-Pulido [3,5], Juan A. Gómez-Pulido [6], Miguel Vargas-Lombardo [4,*], José L. Castillo-Sequera [3,5], Miguel Pablo Aguirre [7], José Sanz-Moreno [1], María-Luz Polo-Luque [5,8] and Diego Rodríguez-Puyol [9]

1 Foundation for Biomedical Research, Hospital Príncipe de Asturias, 28805 Alcalá de Henares, Spain; alberto.garces@ufv.es (A.G.-J.); j.sanz@uah.es (J.S.-M.)
2 Center for Studies and Innovation in Knowledge Management, Universidad Francisco de Vitoria, 28223 Madrid, Spain
3 Department of Computer Science, Universidad de Alcalá, 28805 Alcalá de Henares, Spain; huriviades.calderon@utp.ac.pa (H.C.-G.); jose.gomez@uah.es (J.M.G.-P.); jluis.castillo@uah.es (J.L.C.-S.)
4 E-Health and Supercomputing Research Group, Technological University of Panama, Panama City 0819-07289, Panama
5 Ramón y Cajal Institute for Health Research, 28034 Madrid, Spain; mariluz.polo@uah.es
6 Department of Technologies of Computers and Communications, Universidad de Extremadura, 10003 Cáceres, Spain; jangomez@unex.es
7 Department of Electrical and Electronic Engineering, Technological Institute of Buenos Aires, Buenos Aires C1437FBG, Argentina; maguir@itba.edu.ar
8 Department of Nursing and Physiotherapy, Universidad de Alcalá, 28805 Alcalá de Henares, Spain
9 Department of Medicine and Medical Specialties, Research Foundation of the University Hospital Príncipe de Asturias, IRYCIS, Universidad de Alcalá, 28805 Alcalá de Henares, Spain; diego.rodriguez@uah.es
* Correspondence: miguel.vargas@utp.ac.pa

check for updates

Citation: Garcés-Jiménez, A.; Calderón-Gómez, H.; Gómez-Pulido, J.M.; Gómez-Pulido, J.A.; Vargas-Lombardo, M.; Castillo-Sequera, J.L.; Aguirre, M.P.; Sanz-Moreno, J.; Polo-Luque, M.-L.; Rodríguez-Puyol, D. Medical Prognosis of Infectious Diseases in Nursing Homes by Applying Machine Learning on Clinical Data Collected in Cloud Microservices. *Int. J. Environ. Res. Public Health* **2021**, *18*, 13278. https://doi.org/10.3390/ijerph182413278

Academic Editor: Keun Ho Ryu

Received: 4 November 2021
Accepted: 12 December 2021
Published: 16 December 2021

Abstract: Background: treating infectious diseases in elderly individuals is difficult; patient referral to emergency services often occurs, since the elderly tend to arrive at consultations with advanced, serious symptoms. Aim: it was hypothesized that anticipating an infectious disease diagnosis by a few days could significantly improve a patient's well-being and reduce the burden on emergency health system services. Methods: vital signs from residents were taken daily and transferred to a database in the cloud. Classifiers were used to recognize patterns in the spatial domain process of the collected data. Doctors reported their diagnoses when any disease presented. A flexible microservice architecture provided access and functionality to the system. Results: combining two different domains, health and technology, is not easy, but the results are encouraging. The classifiers reported good results; the system has been well accepted by medical personnel and is proving to be cost-effective and a good solution to service disadvantaged areas. In this context, this research found the importance of certain clinical variables in the identification of infectious diseases. Conclusions: this work explores how to apply mobile communications, cloud services, and machine learning technology, in order to provide efficient tools for medical staff in nursing homes. The scalable architecture can be extended to big data applications that may extract valuable knowledge patterns for medical research.

Keywords: early diagnosis; infections; patients; machine learning; computer systems; internet use; cloud computing

1. Background and Objectives

The world's older population is growing at a significant rate. Today, 8.5% of the population is aged 65 and over; this will increase to 17% by 2050 [1]. Infectious diseases are common (and serious) in this group, often requiring the use of emergency services, degrading their efficiency, and sometimes overburdening them to the point of collapse [2].

The use of digital technology for health supports a patient-centric approach, based on communication, empathy, and collaboration between patients and practitioners [3]. eHealth provides mobility with portable devices, wearables or smartphones, ubiquitous services, and adaptive resources with the cloud. eHealth research addresses electronic records, computerized order entries, e-prescribing, clinical decision support systems (CDSS), telemedicine, health knowledge management, triage, virtual healthcare teams, and medical mobility services (mHealth). eHealth is supported by advanced IT, such as big data, machine learning, artificial intelligence, cloud services, mobile devices, and the internet of things (IoT) [4–6].

However, technology implementations could clash with practitioners and nurses in some cases, as they entail additional workloads. In addition, eHealth decision support systems do not absolve doctors from their accountability [7]. Data-based systems also face the challenge of scarce and poor-quality data in regard to training the model, sometimes becoming unaffordable due to ignoring procedures [8]. The long learning curve of a new eHealth system slows deployment; changes can also be seen as a threat to existing job conditions.

The hypothesis is that anticipating an infectious disease diagnosis by a few days could improve a patient's well-being, alleviate the health system's resources, and result in relatives having a "better perception". Cost-effectiveness is an important driver for national health systems, i.e., to improve medical services for citizens [9]. The evolution of an infectious process is characterized by changes in vital signs. These data could be used to determine the probability of developing an infectious disease [10].

Machine learning (ML) techniques can be used for predictive modeling [11]. They provide pattern recognition and forecast the evolution of a disease, improving the protocols of control and care. ML is an area of computer science derived from artificial intelligence. It provides satisfactory results in eHealth and medicine [12].

This research presents the application of mobile communications, cloud services, and machine learning technology to provide efficient tools to medical staff in nursing homes in order to predict the development of infectious diseases. The approach taken by our study makes it somewhat different from other proposals, due to its particular characteristics. The patients were not cared for in usual health centers, but were elderly people living in nursing homes. Three particular infectious diseases were considered: acute respiratory, urinary tract, and skin and soft tissue infections; a customized biosensor system was developed for the project; the communications infrastructure for data collection, storage, and analysis ws based on microservices; and machine learning algorithms were integrated into the microservices for prediction purposes.

To this end, vital signs from the residents were taken daily and transferred to a database in the cloud by means of an experimental data capture system. Classifiers were used to recognize patterns in the spatial domain process of the collected data. In this context, this research found the importance of certain clinical variables in the identification of infectious diseases, as we will discuss later. These vital signs were selected because they may change due to the pathophysiological adaptations that take place in infectious diseases. Thus, the infection-related inflammation process, stress induced activation of the sympathetic nervous system, and modifications of the activities of the nuclei that regulate heart and lung functioning may modify body temperature, electrodermal activity, oxygen saturation, heart beat rate, and blood pressure.

2. Related Work

The references listed below were selected in relation to the underlying problem of this study: the possibility of early diagnosis by collecting and processing medical data that were further analyzed by advanced hardware and software architectures and tools.

2.1. Early Diagnosis

Given the importance of anticipating the diagnoses of infectious diseases, it is surprising that this concept remains a novelty. Similar approaches for high-risk and very severe cardiopathies [13] or pneumopathies [14] have been proposed. However, anticipating respiratory and urinary infections has not yet been practicable or sustainable, even after considering the high prevalence. The selected infectious diseases based on their prevalence in elderly individuals, are acute respiratory infection (ARI), urinary tract infection (UTI), and skin and soft tissue infection (SSTI).

2.2. Collecting and Processing Medical Data

Sensors used to capture medical signs can be placed in public or private spaces, such as cameras, barometers, microphones, passive infrared (PIR), ultrasound motion detectors, or radio frequency identification (RFID). They can also be embedded in mobile devices, such as accelerometers, magnetometers, or gyroscopes (A/M/G), and be worn, such as smartwatches or alert necklaces [15]. The last category of sensors are those expressly applied on the body of the patient, such as body thermometers, pulse oximeters, tensiometers, electrocardiograms (ECGs), or electroencephalograms (EEGs). The sensors can be networked through wireless sensor networks (WSNs) or body area sensor networks (BASNs).

Once heterogeneous clinical data are collected, they are cleaned, filtered, individualized, and combined. The use of ML, deep learning (DL), artificial intelligence (AI), or ambient intelligence (AmI) in this context [16] motivates research on the automatic identification of the basic activities of daily living. Service provisioning requires self-contention to ensure nonintrusive technology [17].

The information gathered by large sensor networks, such as the IoT [18], makes the utilization of multi-agent management integration [19] and pervasive mobile communications [20], known as mHealth in the medical field, advisable. This allows for delivering new advanced services, such as ECG wearable devices [21], accepted portable mobile applications [22], mobile advisors for drug dosage and adverse reactions [23], medical recommenders for different medical specialties [24–26], fast automatic triage [27], professional medical education programs [28], and telecare systems [29], among many others.

2.3. Hardware and Software Architectures and Tools

The cloud supplies computing power and data storage on demand as scalable commodities under three layers of service: infrastructure (IaaS), platform (PaaS), and software (SaaS). Security and privacy are key requirements for the cloud due to the sensitive information in medical records [5,30].

Microservice architecture is derived from service-oriented architecture (SOA) [31] to provide flexible and scalable execution properties in the cloud. Each microservice plays specific roles depending on the database (DB) requirements [32]. External configuration, microservice discovery, load balancing, central login, metrics, or autoscaling require attention. There are powerful software tools available for delivering applications rapidly by adopting the microservice paradigm [33].

With regard to software tools, ML provides a good approach for data-based predictive analysis [11]. Many algorithms provide knowledge pattern recognition and forecasting, suggesting the possibility of anticipating the diagnosis of diseases from monitored medical data. ML has techniques that have been widely applied, with satisfactory results in eHealth [12].

Some unsupervised learning techniques used in eHealth are K-means, density-based spatial clustering of applications with noise (DBSCAN), self-organized maps (SOMS), similarity network fusion (SNF), perturbation clustering for data integration and disease subtyping (PINS), and cancer integration via multikernel learning (CIMLR), among others. Common supervised learning algorithms in this domain are support vector machine (SVM), iterative dichotomizer 3 (ID3), K-nearest neighbor (KNN), Naive Bayes (NB), Bayesian networks, linear regression, and logistic regression for classification [34].

3. Materials and Methods

This research provides a complete experimental system for gathering medical information from elderly individuals, analyzing the data with predictive ML models.

The experiments performed follow a holistic view, where the data gathering requires defining the scope of the experiment, the protocol or workflow for the medical personnel, the mobile set of instruments, the software platform, and the data analysis techniques.

3.1. Participants, Procedure, and Ethical Considerations

This proposal is intended to be applied for institutional residents in nursing homes that are susceptible to developing infectious diseases. A workflow is defined for nurses to collect vital signs on a daily basis from their assigned residents. Doctors must report any infectious disease detected in any of these residents.

The protocol requires that residents or relatives approve its use beforehand due to normative restrictions and to protect the individual's privacy.

The assigned nurses must follow the following protocol: seated in front of the resident, they (1) turn on the app (on the tablet); (2) switch the hub on, as well as the app connection; (3) select the patient's ID; (4) deploy the medical sensors on the arms and hands; (5) press the start button to start the readings; (6) save the data locally once the measurements are verified; (7) and upload the data to the Cloud DB as soon as the tablet is connected to the internet (WiFi or telephony). Finally, nurses place the devices back into the cases for the next residents. Figure 1 shows the workflow of the procedure.

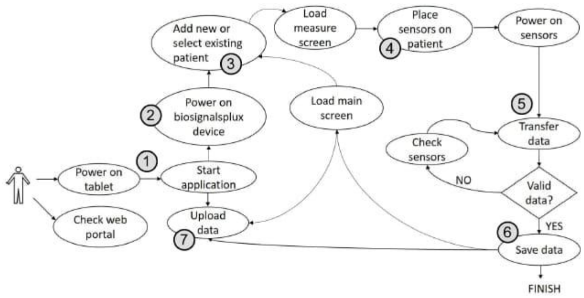

Figure 1. Protocol workflow.

The nurses are to ensure that the batteries of the active sensors, hub, and tablet, are fully charged for the next day, and to promptly report any incidents observed that could compromise the data acquisition.

Finally, ethical consideration for setting clear limits for the research and protecting people's privacy was implemented at the beginning of this project, following the instructions of the founders by means of private statements in their hands.

3.2. Instruments

The nurses use a portable set of biosensors to take the required medical signs from the residents. The equipment must be comfortable to carry and fast to deploy. The different components must be resistant to manipulation, and the application must be robust to perturbances and anomalous events. For general deployment, the equipment must receive approval from the respective health systems.

It is necessary that the equipment be seen as comfortable to operate by the assigned personnel and allow for a warm relationship with the resident. The mobile application that operates the equipment must be fully functional online and offline, avoiding delays in the process.

The field equipment used by the nurses are small cases with customized sets of four sensors obtained from a commercial vendor [35], prepared for the devices and the tablets used for the mobile applications, as shown in Figure 2.

Figure 2. Customized briefcase to carry the medical sensor set.

The four biosensors collected five vital signs, taking (1) the average, maximum, and minimum values of the electrodermal activity (EDA) and heart beat rate; (2) the maximum and minimum oxygen saturation (SPO2); (3) the body temperature when stable; and (4) both the systolic and diastolic blood pressure. A hub quantifies and multiplexes the signals and sends them as a mobile application (app) via Bluetooth, as shown in Figure 3.

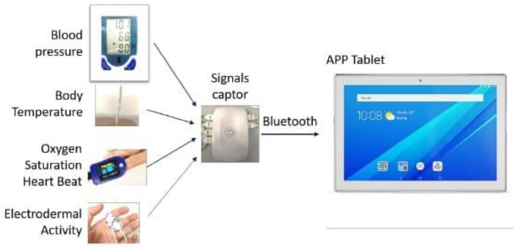

Figure 3. Medical sensor, hub, and android tablet connections.

After preprocessing the signals, the android-based apps check that the values are within the expected ranges, store them in the tablet, and try to connect to the internet when WiFi is available to upload the outstanding records asynchronously to the cloud DB. The operation does not stop when there is no internet coverage. If there are 19 variables (means, dates, flags, etc.), and the device needs to reserve memory, considering the same size for all residents (double precision), and each device is used with 100 residents monitored daily, the local memory required over 2 years would be 84 MB. The current version of the app requires 50 M, which is far less the resources of the tablet, i.e., 11 GB [36], or any current mobile phone.

The hub autonomy is 10 h in streaming mode, and the tablet has a 5000 mAh battery that lasts approximately 3 h, which is enough for one day if it is fully charged overnight. Cables are a problem, however, and the main cause of failure, as they are too thin, and the nurses are in a hurry. Wireless connections for the sensors should work better.

3.3. System Design

This project requires special attention to the software platform, implemented by cloud services and microservice architecture. The software platform must provide flexibility in computation resources and functionality to quickly adopt new services and applications.

Collaboration among different teams must be ubiquitous, and the nature of the data leads to a need for special attention to its security. The cloud suits the requirements well, but it is necessary to respond to security, availability, maintainability, and normative issues [37].

Two main scenarios must inspire the deployment of the software platform: (1) for offline applications, such as medical decision assistants, and (2) for online applications, such as remote real-time monitoring telecare, which encompass the need for immediate actions triggered by the alerts [38].

The data stored in the mobile application are exported in CSV format to a database in the cloud and then remain available for the micro-service-based data analysis tasks.

The SaaS database is asynchronously nurtured from quasi-unlimited concurrent uploading sessions set with each mobile application. A small cloud storage package of 100 GB would support 1190 mobile applications, monitoring 100 residents each, uploading data for 2 years, yielding a total of 119,000 residents. The cloud provides [39] (1) efficient multitenancy; (2) sharing data for different goals; (3) elastic scalability, allowing rapid deployments of complete scenarios, in nonstop ongoing synchronization and self-adapting to the new demands of resources; and (4) data privacy.

The cloud software [40] is implemented in a layered microservice architecture [41], because of [42] (1) the low-cost implementation; (2) the options available to develop different levels of software quality; (3) the scalable and adaptable resource configuration on demand—each component can be individually duplicated; and (4) the compatibility with smart devices and several communication protocols. Mobile devices access the API gateway and user PCs with the web UI. Users call the microservices with their credentials ("nurse", "doctor", or "administrator"). The microservices interconnect themselves to build a single application for users.

The microservices are classified into three groups, all sharing the data via JSON request/response HTTP for (1) interfacing with the physical biosensor application; (2) managing the access policies; and (3) synchronously recording the reports by doctors about patients developing infectious diseases. The architecture uses the representational state transfer (REST) API for data integration, transference, and storage [43]. The software applications are split into small purpose-specific programs with UIs for different domains or APIs to interconnect with third-party applications [44] throughout the infrastructure layer [32], as shown in Figure 4.

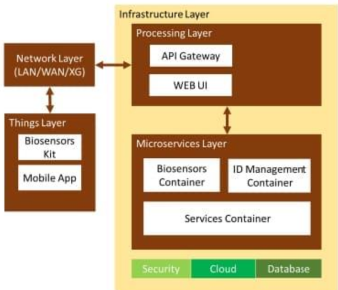

Figure 4. Microservices software architecture for anticipated diagnosis of infectious diseases.

The "things" layer connects the hardware to the network layer and validates the signals. The network layer opens connections to mobile applications for asynchronous data transference [45]. The processing layer assigns each device the corresponding access privileges for their methods to the microservices. The microservices layer provides the service respontses to the specific queries. The infrastructure layer delivers availability, scalability, and data integrity for the upper layers, with networking, processing, and storage resources [46]. It manages data with MapReduce in distributed computing [44] and stores data with Apache Hadoop in Cassandra NoSQL DB [47] to improve the error tolerance. This layer also accepts complex computation tasks from upper layers—authentication or interoperability—to alleviate their resources for their primary operations [48].

3.4. Analysis by Machine Learning Classifiers

The daily medical data gathered from residents can be treated with ML techniques in time or spatial domains, also known as longitudinal or cross-sectional studies. The time approach studies the evolution of certain variables using time series, while the spatial approach considers the relations among the medical variables of a sample, finding patterns and making classifications, according to all samples. Time series are good for predicting, but are not essential. In fact, the spatial domain is more precise for pattern recognition analysis. The spatial domain works complementarily with the time domain approach.

In this work, we approached the analysis of the clinical data under a spatial approach. The absence of the time component prevents a predictive approach, but the spatial dimension allows for a more precise pattern recognition analysis. In this way, it is possible to classify a sample measured from an individual as a sample recognized as possibly indicating an infectious disease. In addition, the information obtained with this approach will allow us to perform a better predictive analysis based on time series in the future.

The learning phase allows us to know the relative weight of each direct or transformed feature in the classification or prediction of an infectious disease. The performance of the classifier can be measured with error performance and the coefficient of determination (R^2).

This analysis is coded with an API of the Waikato Environment for Knowledge Analysis (WEKA) [49], Java-based ML software implemented in the Apache Spark development environment [50], Java Servlet (JS), and Java Server Pages (JSP) [51] for handling the data entries of the microservices. The Java web service is in Apache Maven [52].

The goal of applying these ML methods was not to compare the prediction rate with the true incidence, but to be a first approach for data classification integrated on the web service, to experiment with the influence of the clinical variables on the predictable yielding success with regard to each type of infection.

4. Results

Spanish health authorities approved running this project in two nursing homes in Madrid. The main population and resources of these institutions are shown in Table 1. The data in the table refer to residents participating, where the inclusion criterion was the ability to understand the purpose of the experiment and volunteering.

Table 1. Monitored population and resources.

Population	Cardenal Cisneros	Francisco de Vitoria	Total
Residents	127	316	443
Participants	20	40	60
Participants (%)	16%	13%	14%
Participants who developed disease	7	33	40
Minimum age	79	67	67
Maximum age	94	101	101
Average age	88.7	89.7	89.5
Std. deviation	5.1	7.0	6.6
Medical staff	4	14	18
Start collecting	24 March 2018	4 April 2018	
End collecting	11 March 2019	11 March 2019	

The medical team selected the variables listed in Table 2 and indicated their expected ranges.

Table 2. Vital signs to monitor and life-compatible ranges.

Vital Sign	Valid Range	Out of Range
Body temperature (T)	34 °C < T < 42 °C	T < 34 °C, T > 42 °C
Electrodermal activity (EDA)	EDA > 0.2 μS	EDA < 0.2 μS
Oxygen saturation (SPO2)	70% < SPO2 < 100%	SPO2 < 70%
Heart Rate (HBR)	HBR > 30 bpm	HBR < 30 bpm
Blood pressure (DIA)	DIA > 30 mmHg	DIA < 30 mmHg
Blood pressure (SYS)	SYS > 60 mmHg	SYS < 60 mmHg

The mobile application attaches the date and time of the sample and the patient identification code to the collection of signals; this is manually anonymized by the nurse assigning the code, and the nurse sets flags to indicate if the record has been successfully uploaded.

4.1. Protocol and Acceptance

The healthcare personnel (doctors and nurses) are trained to know the process, look after the equipment, and fix minor incidents. The learning process was conducted with 18 volunteers until they could proceed autonomously. After that, they were requested to simulate, more than once, the process for taking samples and recording the duration. The purpose of this small experiment was to improve the data collection protocol, not only to facilitate the work of the personnel, but also to increase the potential number of residents susceptible to monitoring. Table 3 shows the results.

Table 3. Time (h:min:s) required for training and taking one sample.

Activity	Mean	Std. Deviation
Learning process	0:07:00	0:02:10
Process execution:		
Sensors deployment on the body	0:01:55	0:00:49
APP initialization	0:00:33	0:00:38
Sensors delay	0:01:12	0:00:33
Upload the data to the cloud and resume	0:00:35	0:00:22
Total time consumed per resident	0:04:15	0:01:14

Learning the process takes only 7 min on average. The vital sign collection, on the other hand, only takes 4 min and 15 s on average, yielding the possibility for one nurse to monitor 54 patients in 4 h. Greetings and moving to the next room, along with any other activity could slow down that rate, although practice would compensate and speed up the process.

The experiment also recorded the volunteers' ages, ranging from 20 to 70 years, and digital competency with a self-graded scale from 1 (IT illiterate) to 5 (digital native). Older volunteers spent slightly more time learning than younger volunteers, but the Kruskal–Wallis (KWT) [53] test gave a p-value of 0.428, showing that this result is not conclusive. The KWT—applied for the need to be IT skilled—gave a p-value of 0.088, confirming that the personnel do not need to be IT literate.

The process is fast and comfortable, as only the arms and hands of the residents are exposed to the daily test, not requiring undressing or intimate contact. Additionally, the presence of the nurse helps to generate a friendly environment for caregivers and residents.

The protocol for the doctors does not require anything other than reporting when the resident is developing a disease, which infection the symptoms are compatible with, the date of the alert, and if there was a referral to the emergency services.

4.2. System Efficiency

The software architecture attempts to check its efficiency, integration, compatibility, and performance, counting errors when the components interact [54]. The testbed has two Docker containers (1 vCPU, 4 GB, 120 GB Disk, Ubuntu 16.04.6 LTS); Apache for the endpoint and Nginx as the frontend proxy and load balancer. Nginx receives the user queries and forwards them to the microservices; Apache JMeter measures the workload [55]. BlazeMeter servers (US East—Virginia, AWS) simulate the workload [54,56]. Microservice #1 (MS1) provides a search of over 6297 records, and Microservice #2 (MS2) requests 138 records of residents with infectious diseases for 20 and 50 concurrent virtual users. ALL is the sum of MS1 and MS2. There are two scenarios: EIM-1-FB with frontend–backend "full stack", shown in Table 4, and EIM-2-F, frontend-only, shown in Table 5 and two workload tests of 20 min each with 20 and 50 virtual users (threads), respectively, which gradually increase the queries per second, reaching 14,602 on average.

Table 4. Workload testing report for EIM-1-FB 20/50. The time measurement is the average in ms.

Label	Samples	Resp. Time	Avg. Hit/s	90% Line	99% Line	#Error	Avg. Latency	Users
ALL	29,939	782	25	755	5247	0	252	20
MS1	14,977	811	13	767	5151	0	280	20
MS2	14,962	752	13	719	5311	0	223	20
ALL	29,227	2003	24	1863	13,759	0	582	50
MS1	14,625	1991	12	1863	13,439	0	627	50
MS2	14,602	2016	12	1871	14,079	0	538	50

Table 5. Workload testing report for EIM-2-FB 20/50. The time measurement is the average in ms.

Label	Samples	Resp. Time	Avg. Hit/s	90% Line	99% Line	#Error	Avg. Latency	Users
ALL	3456	6785	3	15,487	38,911	6	3520	20
MS1	1733	6777	1	15,295	37,375	5	3645	20
MS2	1723	6794	1	15,487	40,703	1	3582	20
ALL	215,825	164	180	53	1047	79,944	85	50
MS1	107,925	147	90	56	1047	39,973	90	50
MS2	107,900	182	90	49	1047	39,971	79	50

4.3. Data Analysis

This research analyzed the performance of three space-based ML algorithms: (1) naive Bayes (NB) [56], which is easy to implement and is widely used for medical diagnosis and disease prediction; (2) filtered classifier (FC) [57], which classifies previously filtered or preprocessed data; and (3) random forest (RF) [58], a supervised classifier that is used for this work the random tree (RT) technique to build decision trees for the classification. The reason for choosing these three algorithms is that they allow modifications of the weight of the attributes. In this way, the medical personnel can assign different significance levels to the clinical parameters for diagnostic purposes; for example, we can assign more relevance to body temperature in the classification and analyze how the corresponding results improve (or not) in other cases. The basic settings of these algorithms were those provided by default in WEKA to handle problems of similar complexity.

The analysis starts with a web service for NB and FC as a first approach for data classification, and then a software application to experiment with the weights of the variables for the RF is developed. The next figures help to visualize the data of patients developing infections. Figure 5 depicts the vital signs of one resident developing IRA, Figure 6, one developing a UTI and Figure 7, another developing an SSTI.

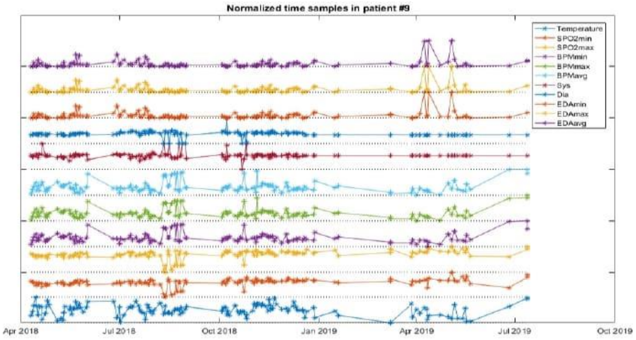

Figure 5. Vital signs evolution of one resident with ARI.

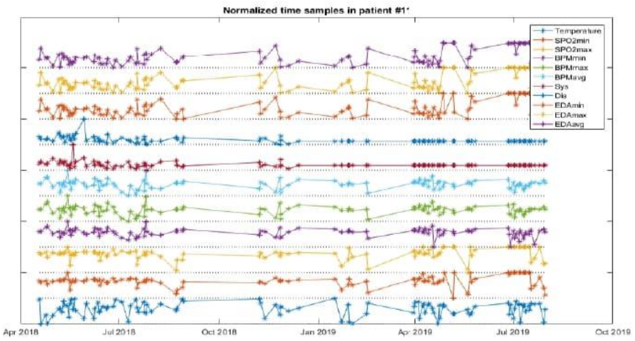

Figure 6. Vital signs evolution of one resident with UTI.

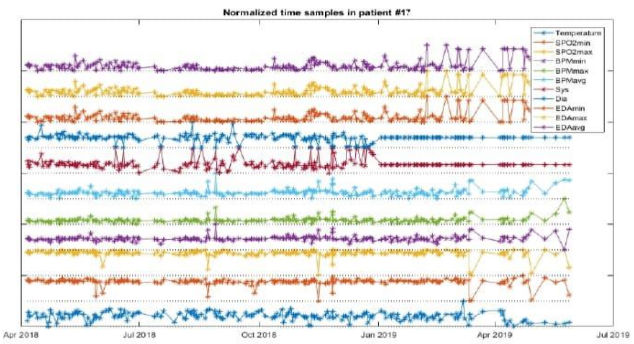

Figure 7. Vital signs evolution of one resident with SSTI.

The WEB service for NB and FC classifies an input sample as a possible ARI, UTI, or SSTI infection. First, the user selects the algorithm and then uploads the medical data record of any resident. The trained model classifies the test data into the three types of infection, providing, as a result, the success rate of the classification (% for the patient developing any of the infectious processes and identifying which one it is) and the weight of each medical parameter affecting the prediction. The model has been trained with other residents who have developed these diseases.

5. Discussion

The obtained results are quite satisfactory, but they are not accurate enough due to the short period for sampling. To obtain more accuracy, it is necessary to either extend the period of the experiment or apply certain techniques for small datasets, such as including context variables, such as the season, holidays, and weather [59].

With RF, the type of infectious disease is deduced from the independent variables, considering low entropy, a measure of the amount of possible information disorder or randomness. All available samples (6277) from all patients (60) are split 95% for the training dataset (5963) and 5% for the validation dataset (314). Only 129 developed ARI, 95 UTIs, and 90 SSTI infections. The process is repeated up to 11 times, changing the weights of the variables according to their importance in the classification. Thus, it is possible to determine which of the 11 variables is better for medical personnel to concentrate on when monitoring the patients. With the available data, the minimum heartbeat rate (HBR) provided the best success rate (%), followed by the average HBR, as shown in Figure 8.

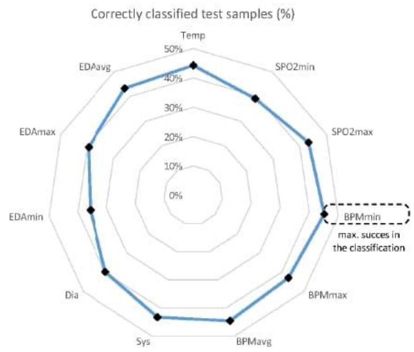

Figure 8. Relative frequency of correct classifications per variable.

Although these figures do not show an impressive performance prediction in general, the results present a significant difference in predictability for each infection. Figure 9 shows the true positives (%) with respect to the variables for each of the diseases.

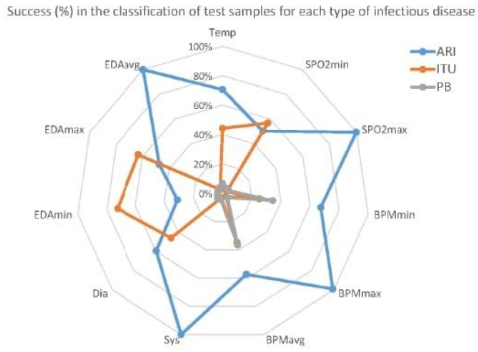

Figure 9. Success ratio detecting infectious diseases. (a) Orange: ITU, (b) Gary: PB, (c) Blue: ARI.

ARI infection is significantly more predictable, yielding 100% success for some variables. On the other hand, SSTI infections are harder to predict with the selected variables. It is necessary to note that there are fewer SSTI training records than for ARI, and the former model was less trained than the latter, again indicating the need to extend the period of sampling.

Finally, the most important variable for all of the diseases is the average HBR.

6. Conclusions

This research proposes a comfortable, flexible, accessible, and cost-effective eHealth monitoring system for residents in nursing homes, and analyzes the predictability of infectious diseases based on the vital signs collected in the pilot study. Its cost-effective implementation allows disadvantaged areas and less accessible populations to be reached.

This paper has demonstrated that the system is easy to use and that there is no need for IT skills for nurses. In addition, the protocol is especially resident-friendly, improving relationships with caregivers.

The microservice-implemented architecture is cost-effective and scalable. The stress tests indicate when the system experiences saturation. The functionality of the microservices to carry out the optimal service has been tested. The mobile application is an open standard that can be installed in any android device with minimum changes.

The problems of the cable organization are under research, with a new design of wireless biosensors ported in a suitcase or a tray. The active elements will be charged on a contactless surface, releasing the nurses or replacing the exhausted batteries. The equipment will be approved by the authorities.

Regarding the data analysis, the main contribution of this work lies in the findings of the importance of certain clinical variables in the identification of infectious diseases. There was no variable with significant relevance for predicting all of the selected infectious diseases. However, among these diseases, the HBR showed a higher impact in the classification. Moreover, the ARI was proven to be better detected than the others. The size of the scenario conditions the classification results. Other factors that affect the predictions are the repeatability of the biosensors, the awareness of the medical personnel, and the type of infection (among other factors). However, this study shows some valuable results and is a good starting point for further research.

Other space- and time-based machine learning techniques will follow this study. In particular,it is interesting to tackle the clinical data analysis under a time series approach. Moreover, larger datasets should be collected to improve the classifier training and the time series accuracy.

Author Contributions: Conceptualization, D.R.-P.; data curation, H.C.-G., J.M.G.-P., M.V.-L., J.L.C.-S., M.P.A., J.S.-M. and M.-L.P.-L.; formal analysis, A.G.-J., J.M.G.-P. and J.A.G.-P.; funding acquisition, J.A.G.-P., M.V.-L. and J.S.-M.; investigation, A.G.-J., H.C.-G., J.M.G.-P. and J.A.G.-P.; methodology, A.G.-J., J.M.G.-P., J.A.G.-P. and J.S.-M.; project administration, J.M.G.-P., J.A.G.-P., M.V.-L., M.P.A. and J.S.-M.; resources, H.C.-G., J.M.G.-P., M.V.-L., J.L.C.-S. and M.P.A.; software, H.C.-G. and J.L.C.-S.; Supervision, J.A.G.-P., M.V.-L., M.P.A., J.S.-M. and M.-L.P.-L.; validation, M.P.A.; writing—review and editing, D.R.-P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the EU Member States and Associated Countries and the Community of Latin American and Caribbean States (LAC) within the 7th Framework Programme for Research and Technology Development (FP7-INCO) grant number ELAC2015/T09-0819 (project SPIDEP: "Design and implementation of a low-cost smart system for pre-diagnosis and telecare of infectious diseases in elderly people") and by the Government of Extremadura (Spain) grant number IB16002.

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to methodology approved by the research project.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Acknowledgments: The authors would like to extend special thanks to all of the clinical and health-care staff of the nursing homes for the elderly, the Francisco de Vitoria and Cisneros centers of the Department of Social Policies and Family of the Community of Madrid in Spain, and Ntra. Sra del Carmen and Carls George in the Dominican Republic, who generously worked with us to obtain the medical data of the residents. Furthermore, the authors acknowledge the support given by Carlos III Health Institute, Spain (ISCIII), Fundación para la Investigación Biomédica del Hospital Universitario Príncipe de Asturias (FIBHUPA), and the National Secretariat of Science, Technology and Innovation of Panama (SENACYT), under the National Research System (SNI).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
AmI	Ambient Intelligence
ARI	Acute Respiratory Infection
BADL	Basic Activities of Daily Living
BASN	Body Area Sensor Networks
CDSS	Clinical Decision Support Systems
CIMLR	Cancer Integration via Multi-kernel Learning
DB	Database
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DIA	Diastolic Blood Pressure
DL	Deep Learning
ECG	Electrocardiogram
EDA	Electrodermal Activity
EEG	Electroencephalogram
FC	Filtered Classifier
HBR	Heart Beat Rate
IaaS	Infrastructure as a Service
IADL	Instrumental Activities of Daily Living
ID3	Iterative Dichotomizer 3
JS	Java Servlet
JSP	Java Server Pages
KNN	K-Nearest-Neighbor
KWT	Kruskal–Wallis
mHealth	Medical Mobility Services
ML	Machine Learning
MSE	Mean Squared Error
NB	Naive Bayes
NIST	National Institute of Standards and Technology
PaaS	Platform as a Service
PINS	Perturbation Clustering Approach for Data Integration and Disease Subtyping
PIR	Passive Infrared
REST	Representational State Transfer
RF	Random Forest
RFID	Radio Frequency Identification
SaaS	Software as a Service
SNF	Similarity Network Fusion
SOA	Service Oriented Architecture
SOMS	Self-Organized Maps
SPO2	Oxygen Saturation
SSTI	Skin and Soft Tissue Infection
SVM	Support Vector Machine
SYS	Systolic Blood Pressure
UTI	Urinary Tract Infection

WEKA	Waikato Environment for Knowledge Analysis
WSN	Wireless Sensor Networks

References

1. Kontis, V.; Bennett, J.; Li, G.; Foreman, K.; Ezzati, M. Future life expectancy in 35 industrialised countries: Projections with a Bayesian model ensemble. *Lancet* **2017**, *389*, 1323–1335. [CrossRef]
2. Gardner, G. The use and abuse of the emergency ambulance service: Some of the factors affecting the decision whether to call an emergency ambulance. *Arch. Emerg. Med.* **1990**, *7*, 81–89. [CrossRef] [PubMed]
3. Adaji, A.; Melin, G.; Campbell, R.; Lohse, C.; Westphal, J.; Katzelnick, D. Patient-Centered Medical Home Membership Is Associated with Decreased Hospital Admissions for Emergency Department Behavioral Health Patients. *Popul. Health Manag.* **2017**, *21*, 172–179. [CrossRef]
4. Liu, Y.; Ren, W.; Qiu, Y.; Liu, J.; Yin, P.; Ren, J. The Use of Mobile Phone and Medical Apps among General Practitioners in Hangzhou City, Eastern China. *JMIR MHealth UHealth* **2016**, *4*, e64. [CrossRef] [PubMed]
5. Fan, L.; Buchanan, W.; Thuemmler, C.; Lo, O.; Khedim, A.S.; Uthmani, O.; Lawson, A.; Bell, D. DACAR platform for eHealth services cloud. In Proceedings of the 2011 IEEE 4th International Conference on Cloud Computing, Washington, DC, USA, 4–9 July 2011; pp. 219–226. [CrossRef]
6. Souza, P.; Hochhegger, B.; Cavalcanti da Silveira, J.; Pretto, C.; Moralles, C.; Tronchoni, A. MediCloud—Telemedicine System Based on Software as a Service (SaaS). *J. Telemed. EHealth Russ.* **2016**, *1*, 38–40.
7. Keen, J. Digital health care: Cementing centralisation? *Health Inform. J.* **2014**, *20*, 168–75. [CrossRef] [PubMed]
8. Kargl, F.; Lawrence, E.; Fischer, M.; Lim, Y. Security, Privacy and Legal Issues in Pervasive eHealth Monitoring Systems. In Proceedings of the 2008 7th International Conference on Mobile Business, Barcelona, Spain, 7–8 July 2008; pp. 296–304. [CrossRef]
9. Wishner, J.B.; Solleveld, P.A.; Rudowitz, R.; Paradise, J.; Antonisse, L. A Look at Rural Hospital Closures and Implications for Access to Care: Three Case Studies. *Kais. Comm. Medicaid Uninsured* **2016**, 1–16.
10. Nishiura, H. Early efforts in modeling the incubation period of infectious diseases with an acute course of illness. *Emerg. Themes Epidemiol.* **2007**, *4*, 2. [CrossRef]
11. Alpaydin, E. *Introduction to Machine Learning*, 3rd ed.; Adaptive Computation and Machine Learning; MIT Press: Cambridge, MA, USA, 2014.
12. Deo, R.C. Machine Learning in Medicine. *Circulation* **2015**, *132*, 1920–1930. [CrossRef]
13. Ong, M.; Romano, P.; Edgington, S.; Aronow, H.; Auerbach, A.; Black, J.; Marco, T.; Escarce, J.; Evangelista, L.; Hanna, B.; et al. Effectiveness of Remote Patient Monitoring After Discharge of Hospitalized Patients With Failure. *JAMA Intern. Med.* **2016**, *176*, 310–318. [CrossRef]
14. Chatwin, M.; Hawkins, G.; Panicchia, L.; Woods, A.; Hanak, A.; Lucas, R.; Baker, E.; Ramhamdany, E.; Mann, B.; Riley, J.; et al. Randomised crossover trial of telemonitoring in chronic respiratory patients (TeleCRAFT trial). *Thorax* **2016**, *71*, 305–311. [CrossRef]
15. Cornacchia, M.; Ozcan, K.; Zheng, Y.; Velipasalar, S. A Survey on Activity Detection and Classification Using Wearable Sensors. *IEEE Sens. J.* **2016**, *17*, 386–403. [CrossRef]
16. Esch, J.; Rashidi, P. A Survey on Ambient Intelligence in Healthcare. *Proc. IEEE* **2013**, *101*, 2467–2469. 6654. [CrossRef]
17. Mohammed, F.; Idries, A.; Mohamed, N.; Al-Jaroodi, J.; Jawhar, I. UAVs for smart cities: Opportunities and challenges. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 267–273. [CrossRef]
18. Uddin, M.Z.; Khaksar, W.; Torresen, J. Ambient Sensors for Elderly Care and Independent Living: A Survey. *Sensors* **2018**, *18*, 2027. [CrossRef] [PubMed]
19. Shakshuki, E.; Reid, M. Multi-Agent System Applications in Healthcare: Current Technology and Future Roadmap. *Procedia Comput. Sci.* **2015**, *52*, 252–261. [CrossRef]
20. Kamel Boulos, M.; Brewer, A.; Karimkhani, C.; Buller, D.; Dellavalle, R. Mobile medical and health apps: State of the art, concerns, regulatory control and certification. *Online J. Public Health Inform.* **2014**, *5*, 229. [CrossRef]
21. Rachim, V.P.; Chung, W.Y. Wearable Noncontact Armband for Mobile ECG Monitoring System. *IEEE Trans. Biomed. Circuits Syst.* **2016**, *10*, 1–7. [CrossRef]
22. Uqaili, A. Smartphone Use among Young Doctors and Their Impact on Patients of Liaquat University Hospital Jamshoro. *Imp. J. Interdiscip. Res. (IJIR)* **2017**, *3*, 161–164.
23. Apidi, N.; Murugiah, M.; Muthuveloo, R.; Soh, Y.; Caruso, V.; Patel, R.; Ming, L.C. Mobile Medical Applications for Dosage Recommendation, Drug Adverse Reaction, and Drug Interaction: Review and Comparison. *Ther. Innov. Regul. Sci.* **2017**, *51*, 216847901769626. [CrossRef]
24. Bourouis, A.; Feham, M.; Hossain, M.; Zhang, L. An Intelligent Mobile based Decision Support System for Retinal Disease Diagnosis. *Decis. Support Syst.* **2014**, *59*, 341–350. [CrossRef]

25. Biswas, A.; Roy, R.; Bhattacharyya, S.; Khaneja, D.; Bhattacharya, S.; Mukhopadhyay, J. Android application for therapeutic feed and fluid calculation in neonatal care—A way to fast, accurate and safe health-care delivery. In Proceedings of the 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) Shenzhen, China, 15–18 December 2016; pp. 938–945. [CrossRef]
26. Rozati, H.; Shah, S.; Shah, N. Smartphone Applications for the Clinical Oncologist in UK Practice. *J. Cancer Educ. Off. J. Am. Assoc. Cancer Educ.* **2014**, *30*, 367–373. [CrossRef]
27. Medina, J.; Castillo Martínez, A.; Gutiérrez Escolar, A.; de Marcos, L.; Sanchez-Pardo, L. Framework for development triages through mobile applications. In Proceedings of the 24thInternational Conference on Information Systems Development Harbin, China, 25–27 August 2015.
28. Franko, O.; Tirrell, T. Smartphone App Use Among Medical Providers in ACGME Training Programs. *J. Med Syst.* **2011**, *36*, 3135–3139. [CrossRef]
29. Muessig, K.; Pike, E.; LeGrand, S.; Hightow-Weidman, L. Mobile Phone Applications for the Care and Prevention of HIV and Other Sexually Transmitted Diseases: A Review. *J. Med Internet Res.* **2013**, *15*, e1. [CrossRef] [PubMed]
30. Zhang, R.; Liu, L. Security Models and Requirements for Healthcare Application Clouds. In Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing, Miami, FL, USA, 5–10 July 2010; pp. 268–275. [CrossRef]
31. Newman, S. *Building Microservices: Designing Fine-Grained Systems*, 1st ed.; O'Reilly Media, Sebastopol, CA, USA: 2015; p. 280.
32. Francesco, P.; Lago, P.; Malavolta, I. Architecting with Microservices: A Systematic Mapping Study. *J. Syst. Softw.* **2019**, *150*, 77–97. [CrossRef]
33. Trihinas, D.; Tryfonos, A.; Dikaiakos, M.D.; Pallis, G. DevOps as a Service: Pushing the Boundaries of Microservice Adoption. *IEEE Internet Comput.* **2018**, *22*, 65–71. [CrossRef]
34. Caballé, N.; Castillo-Sequera, J.; Gomez-Pulido, J.A.; Gómez, J.; Polo-Luque, M. Machine Learning Applied to Diagnosis of Human Diseases: A Systematic Review. *Appl. Sci.* **2020**, *10*, 5135. [CrossRef]
35. Biosignalplux. 2019. Available online: https://www.biosignalsplux.com/en/ (accessed on 17 January 2019).
36. Leotec, "User Manual Version 1.5: Tablet Supernova Qi16 10.1" Quad Core Tablet. 2019. Available online: http://www.leotec.com/ (accessed on Dec. 2019).
37. Dillon, T.; wu, C.; Chang, E. Cloud Computing: Issues and Challenges. In Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, Australia, 20–23 April 2010; pp. 27–33. [CrossRef]
38. Wu, P.L.; Nam, M.Y.; Choi, J.; Kirlik, A.; Sha, L.; Berlin, R. Supporting Emergency Medical Care Teams with an Integrated Status Display Providing Real-Time Access to Medical Best Practices, Workflow Tracking, and Patient Data. *J. Med. Syst.* **2017**, *41*, 1–19. [CrossRef] [PubMed]
39. Curino, C.; Jones, E.; Popa, R.; Malviya, N.; Wu, E.; Madden, S.; Balakrishnan, H.; Zeldovich, N. Relational Cloud: A Database-as-a-Service for the Cloud. In Proceedings of the 5th Biennial Conference on Innovative Data Systems Research (CIDR 2011), Asilomar, CA, USA, 9–12 January 2011; pp. 235–240.
40. Vilaplana, J.; Solsona, F.; Abella, F.; Filgueira, R.; Rius, J. The Cloud Paradigm Applied to e-Health. *BMC Med. Inform. Decis. Mak.* **2013**, *13*, 35. [CrossRef]
41. Manoj Kumar, N.; Mallick, P.K. The Internet of Things: Insights into the building blocks, component interactions, and architecture layers. *Procedia Comput. Sci.* **2018**, *132*, 109–117. [CrossRef]
42. Lewis, G.; Lago, P.; Procaccianti, G. Architecture Strategies for Cyber-Foraging: Preliminary Results from a Systematic Literature Review. In Proceedings of the European Conference on Software Architecture, Vienna, Austria, 25–29 August 2014; pp. 154–169. [CrossRef]
43. Glushkova, D.; Jovanovic, P.; Abelló, A. MapReduce Performance Model for Hadoop 2.x. *Inf. Syst.* **2017**, *79*, 32–43. [CrossRef]
44. Kramer, M.; Frese, S.; Kuijper, A. Implementing secure applications in smart city clouds using microservices. *Future Gener. Comput. Syst.* **2019**, *99*, 308–320. [CrossRef]
45. Simmhan, Y.; Ravindra, P.; Chaturvedi, S.; Hegde, M.; Ballamajalu, R. Towards a data-driven IoT software architecture for smart city utilities. *Softw. Pract. Exp.* **2018**, *48*, 1390–1416. [CrossRef]
46. Choudhary, V. Comparison of Software Quality Under Perpetual Licensing and Software As a Service. *J. Manag. Inf. Syst.* **2007**, *24*, 141–165. [CrossRef]
47. Hernandez, R.; Becerra, Y.; Torres, J.; Ayguadie, E. Automatic Query Driven Data Modelling in Cassandra. *Procedia Comput. Sci.* **2015**, *51*, 2822–2826. [CrossRef]
48. Bello, O.; Zeadally, S. Intelligent Device-to-Device Communication in the Internet of Things. *IEEE Syst. J.* **2014**, *10*, 1–11. [CrossRef]
49. Witten, I.H.; Frank, E.; Hall, M.A. *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed.; Morgan Kaufmann Series in Data Management Systems; Morgan Kaufmann: Amsterdam, The Netherlands, 2011.
50. Zaharia, M.; Xin, R.; Wendell, P.; Das, T.; Armbrust, M.; Dave, A.; Meng, X.; Rosen, J.; Venkataraman, S.; Franklin, M.; et al. Apache spark: A unified engine for big data processing. *Commun. ACM* **2016**, *59*, 56–65. [CrossRef]
51. Poo, D.; Kiong, D.; Ashok, M. *Java Servlets*; Springer: London, UK, 2008; pp. 259–277. [CrossRef]
52. Miller, F.P.; Vandome, A.F.; McBrewster, J. *Apache Maven*; Alpha Press: London, UK, 2010.
53. Kruskal, W.; Wallis, W. Errata: Use of Ranks in One-Criterion Variance Analysis. *J. Am. Stat. Assoc.* **1952**, *47*, 583–621. [CrossRef]

54. Lonetti, F.; Marchetti, E. *Emerging Software Testing Technologies*; Elsevier: Amsterdam, The Netherlands, 2018. 2017.11.003. [CrossRef]
55. Calderón-Gómez, H.; Mendoza-Pitti, L.; Vargas-Lombardo, M.; Gómez, J.; Castillo-Sequera, J.; Sanz-Moreno, J.; Sencion-Martinez, G. Telemonitoring system for infectious disease prediction in elderly people based on a novel microservice architecture. *IEEE Access* **2020**, *8*, 118340–118354. [CrossRef]
56. Murty, M.N.; Devi, V.S. *Pattern Recognition—An Algorithmic Approach*; Undergraduate Topics in Computer Science; Springer: Zürich, Switzerland, 2011.
57. Chandrika, G.; Reddy, E. An Efficient Filtered Classifier for Classification of Unseen Test Data in Text Documents. In Proceedings of the 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, India, 14–16 December 2017; pp. 1–4. [CrossRef]
58. Ho, T. The Random Subspace Method for Constructing Decision Forests. *Pattern Anal. Mach. Intell. IEEE Trans.* **1998**, *20*, 832–844. [CrossRef]
59. Keasberry, J.; Scott, I.; Sullivan, C.; Staib, A.; Ashby, R. Going digital: A narrative overview of the clinical and organisational impacts of eHealth technologies in hospital practice. *Aust. Health Rev. Publ. Aust. Hosp. Assoc.* **2017**, *41*, 646–664. [CrossRef]

2.2 Artículo 2: "Telemonitoring System for Infectious Disease Prediction in Elderly People Based on a Novel Microservice Architecture"

2.2.1 Resumen del artículo

Este artículo describe el diseño, desarrollo e implementación de un conjunto de microservicios basados en una arquitectura que permite la detección y diagnóstico clínico asistido en el ámbito de las enfermedades infecciosas de pacientes de edad avanzada, a través de un sistema de telemonitorización. El sistema propuesto está diseñado para actualizar de forma continua una base de datos médica alimentada con las constantes vitales procedentes de kits de biosensores aplicados diariamente por enfermeras a personas mayores. La base de datos está alojada en la nube y se gestiona mediante una arquitectura de software de microservicios flexible. Los paradigmas computacionales del borde y la nube se utilizaron en la implementación de una arquitectura de nube híbrida con el fin de soportar aplicaciones versátiles de alto rendimiento bajo el patrón de microservicios para el prediagnóstico de enfermedades infecciosas en pacientes de edad avanzada. Los resultados de un análisis de la usabilidad del equipo, el rendimiento de la arquitectura y el concepto de servicio muestran que el sistema de eHealth propuesto es viable e innovador. Los componentes del sistema también se han seleccionado para que su aplicación resulte rentable para las personas que viven en zonas desfavorecidas. El sistema de sanidad electrónica propuesto también es adecuado para la computación distribuida, los macrodatos y las estructuras NoSQL, lo que permite la aplicación inmediata de algoritmos de aprendizaje automático y de IA para descubrir patrones de conocimiento de la población general.

2.2.2 Conclusiones del artículo

En este trabajo se han analizado, descrito y justificado todos los pasos implicados en el diseño, desarrollo e implantación de los servicios de e-salud que componen la plataforma del proyecto SPIDEP. Esta plataforma está basada en una arquitectura de cinco niveles mediante microservicios. El objetivo de la plataforma es crear un marco basado en el uso de las nuevas TIC para apoyar el diagnóstico precoz de las enfermedades infecciosas con los beneficios añadidos de aumentar el nivel de servicio en la atención médica y la reducción de costes. De igual manera, el sistema de eHealth propuesto también es adecuado para la computación distribuida y para el uso de big data y estructuras NoSQL que permiten la aplicación inmediata de algoritmos de aprendizaje automático e IA para descubrir patrones ocultos en los datos de salud de esta población. Pero, sobre todo, la innovación clave que aporta el sistema de telemonitorización es la posibilidad de obtener la información médica necesaria para construir modelos analíticos y predictivos en conjuntos con los nueve microservicios desarrollados. Además, esta propuesta abre el camino a futuras investigaciones para el diseño de servicios, basados en el flujo de trabajo propuesto para la automatización, integración y despliegue continuo de servicios, con el objetivo de conseguir un mayor rendimiento en términos de organización y atención al paciente.

2.2.3 Contribuciones del artículo a la comunidad científica

El segundo artículo aporta una nueva visión en el contexto de la televigilancia médica, ya que se describe una novedosa metodología para el diseño, desarrollo e implementación de los microservicios que conforma a la plataforma SPIDEP, a diferencia con el primer artículo, este se enfoca más en el aspecto de Ingeniería de Software, ya que el doctorando propone un flujo de trabajo más maduro para el despliegue continuo y automatización de los microservicios desarrollados basándose en los puntos débiles de esta tecnología (por ejemplo: no considerar los efectos de rendimiento al descomponer una aplicación en múltiples servicios, no considerar una infraestructura en la nube para el funcionamiento de los microservicios, etc) trae consigo, una correlación negativa entre el aumento de la latencia y la degradación del rendimiento general de la aplicación hasta su colapso e inviabilidad de ser implementado en los sistemas de salud público; mientras que, los coautores aportaron en el sistema de apoyo a la toma de decisiones clínicas y su validación con el equipo médico. Otra forma de contribuir a la comunidad científica es la flexibilidad de nuestra metodología en adaptarse a diferentes escenarios médicos [55].

2.2.4 Identificación del artículo

- Título: "Telemonitoring System for Infectious Disease Prediction in Elderly People Based on a Novel Microservice Architecture"
- DOI: https://doi.org/10.1109/ACCESS.2020.3005638
- Autores: Huriviades Calderón-Gómez; Luis Mendoza-Pittí; Miguel Vargas-Lombardo; José M. Gómez-Pulido; José L. Castillo-Sequera; José Sanz-Moreno; Gloria Sención
- Acceso del artículo: Open Access (OA)
- Fecha de publicación: 29 de junio de 2020
- Idioma: Inglés

2.2.5 Índices de calidad del artículo

- Revista: IEEE Access
- ISSN: 2169-3536
- Editor: IEEE
- Volumen y número: 8-1
- Índice de impacto y ranking JCR (2021): 3.476
    - Computer science, information systems – Q2
- Índice de impacto y ranking SJR (2021): 0.927
    - Computer Science (miscellaneous) – Q1

2.2.6 Artículo publicado

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 1, 2020, accepted June 24, 2020, date of publication June 29, 2020, date of current version July 7, 2020.
Digital Object Identifier 10.1109/ACCESS.2020.3005638

Telemonitoring System for Infectious Disease Prediction in Elderly People Based on a Novel Microservice Architecture

HURIVIADES CALDERÓN-GÓMEZ[1,2], LUIS MENDOZA-PITTÍ[1,2], MIGUEL VARGAS-LOMBARDO[2], (Member, IEEE), JOSÉ MANUEL GÓMEZ-PULIDO[1,3], JOSÉ LUIS CASTILLO-SEQUERA[1,3], JOSÉ SANZ-MORENO[4], AND GLORIA SENCIÓN[5]
[1]Department of Computer Science, University of Alcalá, 28805 Alcalá de Henares, Spain
[2]E-Health and Supercomputing Research Group (GISES), Technological University of Panama, Panama City 0819-07289, Panama
[3]Ramón y Cajal Institute for Health Research (IRYCIS), 28034 Madrid, Spain
[4]Foundation for Biomedical Research, Hospital Universitario Príncipe de Asturias, 28805 Alcalá de Henares, Spain
[5]School of Medicine, Autonomous University of Santo Domingo, Santo Domingo 10105, Dominican Republic

Corresponding author: Miguel Vargas-Lombardo (miguel.vargas@utp.ac.pa)

This work was supported in part by the Ministry of Science, Innovation and Universities of Spain, in part by the Foundation for Biomedical Research of the Prince of Asturias University Hospital (FUHUPA), in part by the National Secretariat of Science, and the Technology and Innovation of Panama (SENACYT-PANAMA) through the National Research System (SNI-PANAMA) [Design and implementation of a low-cost intelligent system for pre-diagnosis and telecare of infectious diseases in older people (SPIDEP)] under Grant AC16/00061, and in part by the 2nd Joint Call for Research and Innovation (ERANet-LAC) of the Seventh Framework Program for Research and Technological Development (FP7) of the European Union under Contract ELAC2015/T09-0819 SPIDEP.

ABSTRACT This article describes the design, development and implementation of a set of microservices based on an architecture that enables detection and assisted clinical diagnosis within the field of infectious diseases of elderly patients, via a telemonitoring system. The proposed system is designed to continuously update a medical database fed with vital signs from biosensor kits applied by nurses to elderly people on a daily basis. The database is hosted in the cloud and is managed by a flexible microservices software architecture. The computational paradigms of the edge and the cloud were used in the implementation of a hybrid cloud architecture in order to support versatile high-performance applications under the microservices pattern for the pre-diagnosis of infectious diseases in elderly patients. The results of an analysis of the usability of the equipment, the performance of the architecture and the service concept show that the proposed e-health system is feasible and innovative. The system components are also selected to give a cost-effective implementation for people living in disadvantaged areas. The proposed e-health system is also suitable for distributed computing, big data and NoSQL structures, thus allowing the immediate application of machine learning and AI algorithms to discover knowledge patterns from the overall population.

INDEX TERMS Artificial intelligence, e-health, elderly people, infectious diseases, microservice architecture, microservices, telemonitoring.

I. INTRODUCTION

It is becoming increasingly difficult to ignore the growth of an aging population in Europe, since a 74% rise in the population aged over 65 is expected by 2060 [1], [2]. This group of people have high rates of comorbidity [3], causing a very high consumption of resources in terms of both primary and specialised care [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Sunil Karamchandani.

A very important problem affecting this group of people is the frequent use of emergency services or prolonged hospital stays, caused mostly by infections [5] and especially respiratory and urinary infections [6]–[8]. This is due to the characteristics of the patients; however, these patients tend to go to the emergency room at relatively advanced stages of disease [9]. In many cases, these patients have non-specific neurological symptoms, general symptoms or other problems that are apparently not related to the infection [7].

In view of this, the diagnosis of infectious diseases constitutes a problem in that limited medical care is available (e.g. areas with difficult access or rural areas), since these patients require special care due to the complexity of clinical management [7], [10]. This is reflected in an increase in the mortality rate of patients due to delays in treatment [8], [11]. A simple solution would be to increase the human resources available to care for the elderly; however, this solution is unsustainable in the current economic environment of the public sector.

For this reason, it is important to solve this problem by employing other sustainable initiatives such as telemonitoring, since this allows for the adequate management of at-risk populations. In the case of at-risk long-term patients, both medical and nursing care are generally available, but infectious diseases are often diagnosed at later stages, when hospitalisation is required. If the vital signs of these at-risk patients are regularly monitored, it is possible to carry out preventive treatments or interventions in order to minimise health problems and decrease emergency assistance [5]. For patients living in environments without easy access to health resources, telemonitoring is a low-cost, effective way of offering preventive treatments for these diseases [12].

A project called "Design and implementation of a low-cost intelligent system for the pre-diagnosis and telecare of infectious diseases in elderly people (SPIDEP)" [13] was carried out by a consortium of Latin American and European R&D entities, the aim of which was to build an intelligent system based on information technologies (ICT) to support the early diagnosis of infectious diseases [14] by integrating a machine learning-based inference system to improve decision support in the prevention, treatment and management of infectious diseases [6], [15].

Special emphasis was placed on the design, development and implementation of the services that make up the SPIDEP platform using the architectural pattern of microservices [5], [14].

This article is part of a broad study in the area of infectious diseases developed jointly with the clinical teams of the Hospital Universitario Principe de Asturias Infectious Diseases Unit (Alcalá de Henares, Spain), Chronic Diseases and Cancer Area of the Ramón y Cajal Institute for Health Research (Madrid, Spain) and School of Medicine Autonomous University of Santo Domingo (Dominican Republic). This study was focused on three target groups (acute respiratory infections, urinary tract infections and skin and soft tissue infections). The article presented is also based on previous works in the computation sciences area realized by the partners of the SPIDEP project [4]–[6], [9], [14].

It should be noted that this study aims to contribute to increasing knowledge in the pre-diagnosis of infectious diseases and offer an alternative vision of telemonitoring of elderly people, with the support of ICTs, by the medical team [5], [6], [9].

This present work aims to incorporate and improve the results obtained from the previous studies that have achieve several significant advances for the medical and health care team, which are: the process of taking measurements is very fast, managing to collect and send the measurement data in less than 2 minutes. These features are essential for the wellness of the resident, who perceives the personal care without discomfort [9] and also the quality of health care grows as well as the satisfaction of residents, relatives and nursing personnel [6]. Another important advantage of the speed of the measurement process is that it increases the productivity of the nursing staff, increasing the residents/elderly assistant ratio [5]. In addition, this benefits the patient directly, since pre-diagnosis through a Clinical Decision Support System (CDSS), supported by telemonitoring, allows the detection of the infection in an initial state and can anticipate the timely administration of corresponding medical treatment, thus avoiding the aggressive spread of the infection afflicting the patient [6], [14].

Nowadays, a group of the authors of this research is adapting the software architecture built through SPIDEP to another infectious disease such as (COVID-19) in the Republic of Panama (funded from the National Secretariat of Science, Technology and Innovation of Panama). The purpose of the project is to determine and establish infection sources (clusters) and store the data in a repository that will store the spatial-temporal information of the infection network, which will make it possible to study the behavior of the disease (COVID-19) during and after the epidemic, relying on the use of the proposed architecture linked to AI algorithms for multivariate statistical analysis of the information collected, whose purpose would be to generate forecast curves with granularity at the regional level to support contingency plans by decision-makers.

This project is being developed with funds from the National Secretariat of Science, Technology and Innovation of Panama (SENACYT-PANAMA).

The present article starts with a brief description of related work, including the motivation for the SPIDEP project and its importance within the field of telemonitoring (e-health). We describe each component of the system in the SPIDEP architecture. Section III presents the results of performance tests (peak tests) at the network and hardware level. The pros and cons of the results are evaluated in Section IV. Finally, Section V presents the conclusions of this study, discusses its potential, and suggests future research activities.

II. RELATED WORK

Microservices represent a relatively new approach to software architectural patterns [16]–[18], as they allow for the development of applications as sets of small services that run independently; in other words, it is not necessary to use the same languages or development platforms [19], [20]. Furthermore, they are interoperable with various communication protocols, since they communicate using lightweight mechanisms such as HTTP [21].

Some research studies have therefore suggested combining this architectural pattern (microservices) with various computational paradigms (cloud, fog or edge) [22], [23] in various health scenarios, with the aim of significantly improving the accuracy of medical diagnosis by establishing predictions based on expert systems or other types of artificial intelligence.

Mendes *et al.* [1] made use of microservices for the remote monitoring of biometrics (blood pressure and electrocardiograms) and environmental data in a domestic environment specifically designed for the elderly population. They chose this architectural pattern in conjunction with the fog paradigm, with the aim of providing a secure and flexible system in which each service scales independently based on the demand for data processing and analysis.

Likewise, Grgurić *et al.* [24] presented a pilot system called SmartHabits, based on existing IoT architecture and microservices practices; this study focused on the provision of an intelligent service that reassures family members that their loved ones (elderly people) are doing well. This system consists of three main modules: a home-sensing platform (located within the home of an older person living alone), the SmartHabits expert system (located in the cloud, and designed to detect anomalous situations within the home and to deliver notifications), and the underlying communication structure. They focused on the aspect of flexibility in order to meet the needs of users (interface), based on the use and management of smart devices at home, such as smartphones, tablets or smart TVs.

In a study by Roca *et al.* [25], a chatbot architecture was proposed for chronic patient support based on three pillars: scalability through microservices; standard data exchange models through HL7 fast healthcare interoperability resources (FHIR); and modelling of standard conversations using AIM. These three pillars relied on a microservices-based logic that aimed to process user information and perform automated tasks to provide scalability in a healthcare chatbot ecosystem.

Semenov *et al.* [26] suggested a FHIR-based microservices platform that integrated hospital information systems and clinical decision support systems into a unified information space using microservices. A set of separate services were developed for the microservices platform, i.e., asynchronous nodes distributed in groups, whose purpose was to ensure the compatibility and interoperability of its services based on 50,000 transactions per day with more than 400 decision support models.

Alvarez *et al.* [27] applied a microservices-based architecture for the implementation of a failure detection and diagnosis scheme for teleoperated knee rehabilitation devices via the internet, in order to facilitate a medical development protocol for the recovery of the patient's mobility in the case of geographical vulnerability. They demonstrated that the proposed architecture increased the efficiency of development, as each component was independently designed, implemented, and validated for heterogeneous systems.

Meanwhile, Andrikos *et al.* [28] developed a system that allowed for real-time advanced teleconsultation services on medical imaging (radiologists). It consisted of three modules: user access control (UAC), user management (UM) and content management (CM), based on the microservices pattern. The focused on persistence of the data, since each service managed its own database, using either different instances of the same database technology or completely different database systems.

Another approach developed by Khoonsari *et al.* [29] described a workflow for the analysis of data from metabolomics, focusing on the performance aspect. Microservices were included in the development of the necessary components in order to allow for the analysis of encapsulated data while providing reproducible data analysis solutions that were easy-to-run and easy to integrate on desktops and on public and private clouds (Docker containers). Their workflow was validated using various types of data; for instance, two mass spectrometry, one nuclear magnetic resonance spectroscopy and one fluxomics study were used to demonstrate that the method applied scaled optimally when more computing resources became available.

The present work provides a new vision of what has already been carried out in the aforementioned studies, since our contribution is oriented towards the context of medical telemonitoring focused on microservices (SPIDEP) within the field of infectious diseases of elderly patients, with the aim of allowing for the design, development and implementation of services based on a layered architecture in microservices [5], [14]. The objective of this proposal is to offer regular remote monitoring of vital signs by medical personnel [9]. Preventive care can minimise the severity of infectious processes, consequently reducing the resources necessary to adequately control the problem [14].

In view of the aforementioned considerations, it is very important to define the functionalities of the proposed software with a focus on addressing the following needs. Firstly, each component must have the ability to replicate to balance the load as required. Secondly, microservices require horizontal scalability, which allows for a faster and more precise reaction to peaks in demand. Finally, it should have the ability to cope with a large volume of medical data generated from the integration, transfer and storage of biometric sensors in SPIDEP.

For the reasons indicated above, versatility and high performance were identified as fundamental characteristics of the SPIDEP project. These characteristics allow the needs of the environment of health organisations to be satisfied via a focus on agile scaling of microservices that are heavily used and replicating them across multiple containers without underutilising computing resources [19], [30].

III. OVERVIEW OF THE SPIDEP ARCHITECTURE

A. DESIGN OF THE ARCHITECTURE

In this work, we used a fusion of computational paradigms (edge and cloud) [31], [32], based on a hybrid cloud architecture, with the aim of supporting the implementation of versatile high-performance applications using the microservices pattern. The main arguments for this approach are as follows:

- Low-cost implementation.
- Several options for developing different levels of software quality.
- Scalable and adaptable configuration of resources on demand, since each component can be individually duplicated.
- Compatible with several smart devices and communication protocols.
- Users can run their applications without requiring control by the host.

The architecture can be divided into five functional layers:

- Things layer: This interacts with the processing layer and the network layer, to connect the hardware and validate the signals.
- Network layer: This sets up connections with each APP for the asynchronous uploading of medical data.
- Processing layer: Microservices in this layer use the representational state transfer (REST) protocol to communicate with each other. Each type of device has a different form of access to call the GET, POST, DELETE and PUT methods. In this approach, smartphones and tablets will enter through an API gateway, and computers will enter through user interfaces (UIs); these include the admin UI for the administrator and the nurse/doctor UI for the medical work unit. Each access will be associated with several microservices.
- Microservices layer: This contains microservices for the end users, such as medical personnel and nursing home administrative staff, with their respective profiles. Users can look up historical records, add new biomedical data, manage user profiles and roles, and access pre-diagnosis services. Although each database intercommunicates via JSON request/response HTTP, each microservice has a unique database allowing for independent operation.
- Infrastructure layer: This supports communications, scalability, availability and data integrity for the upper layers. It provides network, server and storage resources for the external clients that connect with each microservice [33], with the aim of achieving greater tolerance to errors in the cloud environment, for example by using MapReduce to manage data in distributed processing [34] and Apache Hadoop to store the data corresponding to each group of microservices within NoSQL database storage structures (Cassandra) [35]. The infrastructure layer also processes complex computation tasks from the upper layers, such as those relating to authentication or interoperability, by alleviating overloaded public or private cloud resources for other primary functions [30], [36], running on Docker containers [37].

In addition, all the interactions between the services within the architecture are processed by the REST API gateway, which acts as a proxy for the microservices (single-entry point into the platform) [38]. This allows for the management of other functional capabilities, such as caching, token management, and microservices monitoring [38], [39].

Likewise, the interactions of each microservice are independent of the physical hosting, and these are either treated individually or grouped into servers or containers [16]. It is therefore important to consider the interaction model that will be implemented in the application, for example for remote procedure calls (RPCs) and event-based interactions [26], the purpose of which is to guarantee that the architecture is robust against failures due to continuous stress at different layers [21], [40].

B. NEW ASPECTS OF THE ARCHITECTURE

The microservices architecture is [17], [41] is a variant of the service-oriented architecture (SOA), both of which provide flexible and scalable properties for execution in the cloud. Each microservice takes on a specific role depending on the requirements of the database [19], [42], [43]. Research in different areas such as system quality, smart city clouds, migration or mobile-oriented applications [41], [44] make use of this approach. However, microservices have certain challenges; for example, their external configuration, microservice discovery, load balancing, central login/logout and metrics or auto-scaling require attention.

The novelty of our proposal lies in the use of microservices in clinical forecast scenarios (infectious diseases), as we propose a continuous construction of an online database to obtain predictive models for the detection of infectious diseases in elderly patients, inspired by the SPIDEP project [13]. A recommender system [6] is also used to support remote assistance in interpreting changes in the vital signs of institutionalised people and in triggering early alerts in the case of possible infection.

Before considering the design of the different microservices for the development of different medical platforms (such as the SPIDEP platform) based on this architecture, it is important to take into account the hierarchy and consolidation of the clinical information of the patients. This is not only necessary for generating standard reports, but can also influence decision making using parameterised, consistent and verified indicators of the collected data in conjunction with other medical information systems [25], [45], [46] that aim to correctly filter the data to minimise false alarms by the recommender systems.

In our case studies (SPIDEP) [9], the standardisation of the data from different nursing homes is a fundamental process in the development of microservice-oriented applications [14]. For this reason, unique records (unique transaction identifiers) represented in EMR-JSON format were used to allow for the use of different types of databases, depending on the particular storage requirements, and in turn provide interoperability to the system (microservices layer).

It was also necessary to customise the fifth layer of the architecture (infrastructure layer) to support a high level of traffic of clinical data on a daily basis. As a consequence, each microservice has a load balancer (NGINX) to reduce the load for remote calls and in turn to answer users' requests from the application [16]. We therefore use a container orchestrator (Kubernetes) [38] with Docker containers [37].

This customisation means that the architecture is compatible with various cloud computing service providers (Azure, AWS, Google Cloud), in order to run thousands of scalable containers regardless of the infrastructure implemented (private, hybrid or public cloud). In this way, a massive and balanced cluster can be implemented with Kubernetes [38], allowing users to consistently meet the demand for e-health services. The advantages of this management system are numerous; for example, each component can be replicated to achieve load balancing as required [47]; when a small functionality scale is applied, the use of microservices allows the reaction to the peak demand to be take place in real time [38]; the use of containers substantially improves autoscaling, both in terms of the response time of applications and the management of IT resources, which is more rapid and efficient than in virtualised environments (VM) [38], [48]; the use of component modularity makes the system robust to failure [40]; and finally, the decoupling provided by the microservices facilitates its maintainability over time [49].

C. ARCHITECTURE IMPLEMENTATION

1) PLATFORM

As part of the concept of a low-cost platform, this work is based on the proposal of a microservices architecture; this is capable of supporting general service communication with the central cloud environment [9], and in conjunction with a machine learning subsystem can improve decision support for the pre-diagnosis of infectious diseases [6]. Structurally, the SPIDEP platform was developed using microservices architecture (with nine separate services), in which these services work as asynchronous nodes distributed as groups [26], [50], [51] which communicate via the REST communication protocol [52]. It should be noted that the breakdown of the components into small independent services was based on our preliminary proposal for a software architecture for SPIDEP (with five layers), referred to here as version Beta v1 [14].

Three versions of this project were developed: version Alpha (partial implementation of three microservices, with instances using MariaDB) [5], version Beta v1 (implementation of five microservices with instances using a MariaDB Galera cluster) [14], and version Beta v2 (current implementation of the new microservices architecture, with hybrid instances in MariaDB Galera Cluster and NoSQL), with the aims of achieving the characteristics necessary for the implementation of versatile high-performance applications and identifying the key services for the optimal operation of version Beta v1. The existing components were therefore decomposed and restructured along with their data [16], [18], [52].

The flow of the SPIDEP platform in relation to microservices is briefly explained below:

- The scheme in SPIDEP is made up of two end points depending on the device (for mobiles, this is the API gateway, and for PCs the web UI).
- The call to the different microservices corresponding to the credentials is established (by a nurse, doctor, IT developer or administrator).
- Based on the considerations described above, microservices are classified based on the common characteristics of their operation [52], [53] into three groups [14]: the first performs the control and management of user accounts, roles, permissions [54] and manages the measurements received from the biosensors [55], [56]; the second group administers the different patients, affiliated institutions and manages the medical data collected by the first group 14]; and the third is responsible for infectious disease registries. It is also responsible for receiving notifications from the clinical decision support system (CDSS) for the prevention, treatment and management of the three groups of infectious diseases, using the model detailed by Baldominos *et al.* [6]. The proposed platform aims to provide new services and functionalities for the changing needs of users, from the standpoint of management, technology, security and legality [57]. The implementation of the developed architecture provides an ecosystem that is scalable according to demand; however, in order to achieve optimal scalability, it is vital to verify the necessary components (language, database, libraries etc.) for each microservice and the data that will be exchanged between them [29]. Failure to do so would increase network traffic between microservices and HTTP resource APIs (causing saturation), and the overall performance of the application would consequently be degraded until it collapsed [19], as shown in Fig. 1.

For the reasons mentioned above, the architecture implemented in the SPIDEP project was chosen (decompose by verb or use case) [58], because it allows for the creation of flexible and scalable software solutions that run in the cloud [41], [59]. However, the strengths and weaknesses of microservices must also be considered.

The main strengths of the microservices are as follows:

- Microservices can be implemented in a given language and can use the libraries that are best suited to provide the functionality needed [25], [29].
- The structure of the platform allows for different types of databases depending on the requirements 25].
- This allows for the agile scaling of individual microservices, which are widely used by replication based on several containers, without needing to replicate underutilised services [19].
- The automation of the infrastructure allows for a reduction in the manual effort involved in building, and for deploying and operating microservices, thus enabling continuous delivery [21].

However, when implementing microservices, the following weak points or difficulties must be considered:

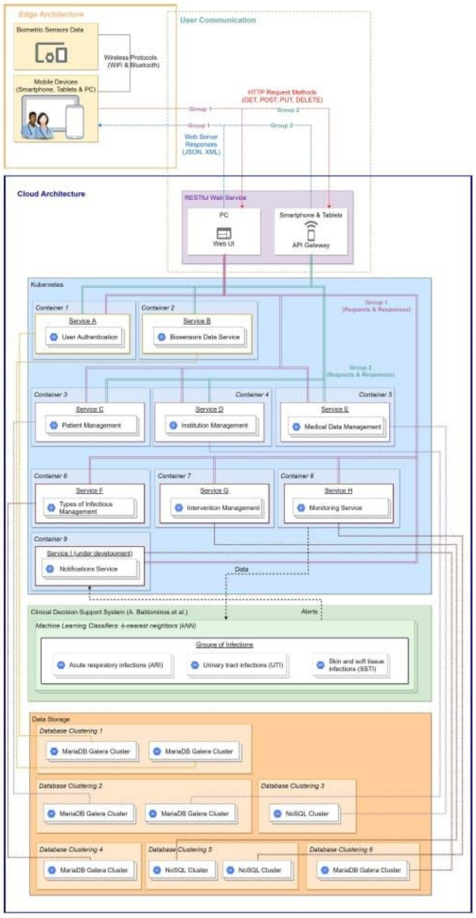

FIGURE 1. General layout of the microservices applied in the SPIDEP platform together with the notification service provided by the CDSS [6].

- This approach requires a clear overview of the data structure and the business processes used in the organisation [40].
- There are several major constraints that directly affect the performance of microservices, including the network, the organisation of complex services, the consistency of data and transaction management at the database level, and the methods of deployment (bare metal server, virtual machine or container). Consequently, cloud infrastructures play a fundamental role in the operation of microservices and their complexity [16], [19], [21].

It should be noted that the user experience in terms of usability of the operation with the different user services was analysed and validated by both the health personnel and the researchers from the international consortium [9], [14]. In addition, the execution and performance of the platform developed in SPIDEP project were shown to meet established standards through various performance tests (spike tests) for each service [42], [59].

2) WORKFLOW FOR THE AUTOMATION OF MICROSERVICES ON THE SPIDEP PLATFORM

Each task involved in an assisted clinical diagnosis in the field of infectious diseases of elderly patients has its own associated microservice. This design allows for the incorporation of new features or the integration of new components developed by third parties.

It is therefore important to define the workflow of the automation for the integration and continuous deployment of services on the SPIDEP platform. This workflow is used by cross-team (typically small groups of six to eight people [60], [61]) for the microservices development process, with the objective of reducing the time between the acceptance of a change into the system and putting it into production [62].

Based on previous research, a proposal was prepared for the automation workflow and was applied to the SPIDEP platform [38], [39], [58], [62]. Fig. 2 shows how this was achieved.

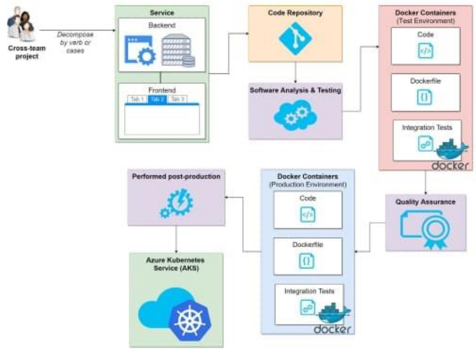

FIGURE 2. Framework for the automation and continuous deployment of services on the SPIDEP platform.

An overview of the proposal is given below.

- Cross-team project: This is responsible for the development, implementation and supervision of the assigned service [63]. It allows for the improvement of health services based on three important characteristics: (i) management, which allows for the creation of personalised interfaces for different groups of users (decision makers, administrators or citizens), according to the requirements and the administrator's credentials; (ii) interoperability, which supports the development of APIs for interconnection with third-party applications or other medical systems; and (iii) versatility, which gives developers the freedom to choose the information technologies (languages, databases, libraries and others) based on which their services will be developed.

- Service: Each team, if required, has the autonomy to decide which is the best technology for the development of their service in order to meet the requirements [63]. The SPIDEP microservices were developed in two programming languages, PHP and Python. The microservices developed in PHP use the Laravel framework, while the those developed in Python use the Django framework. It should be noted that all microservices developed in SPIDEP have an asynchronous HTTPS server based on TLS for client authentication [25], [54], and the databases applied to each one were SQL (MariaDB) and NoSQL (Cassandra or MongoDB). As a complement to this microservices-based architecture, we use the "service instance per container" and "database per service" implementation patterns [64].
- Code repository: A Git repository is used to maintain version control of the code from the development stream of each service [29], [65].
- Software analysis and testing: The source code is reviewed to ensure that it meets the required standard and unit tests are performed on each component [62].
- Docker containers (test environment): The tools necessary to run the software are encapsulated inside a test container (Ubuntu server 18.04 LTS, 1 core, 1.5 GB RAM). These containers must meet preliminary test criteria defined by the developers [65], [66], and are then published on the Docker Hub (private repository) for download and used in SPIDEP in the testing environment.
- Quality assurance (QA): Functional tests, load tests and performance tests are carried out to ensure the quality of the microservice [59], [62].
- Docker containers (production environment): If the microservices pass all the tests defined at the QA stage, the container is deployed to a production instance (Ubuntu server 18.04 LTS 1∼2 Core, 3∼4 GB RAM) [38] [49]. Otherwise, it will be necessary to debug the code and repeat the QA tests. The container is then published to a private repository in the Docker Hub.
- Performed post-production: Additional tests (spike tests) are run to ensure that the new version works properly in the production container [16], [59], [62].
- Azure Kubernetes service (AKS) – Kubernetes is used for container orchestration [38], [67], as it allows for initialisation, scaling of container-based jobs, service exposure, and rescheduling of failed jobs and long-running services [29]. We chose Kubernetes rather than other orchestrators (e.g. Swarm) because it is more widely used in production environments (e.g. Azure) and because of its great versatility in container management [37].

IV. VALIDATION OF RESULTS

It was necessary to evaluate the performance of the microservices developed for the SPIDEP project through an analysis of the proposal. We therefore used performance tests (spike tests) and obtained quantitative values that could be used to measure the overall performance of the application and its interaction with end users. [16], [42], [59], [68], [69].

For this purpose, we used two AKS-controlled testing environments consisting of nine Docker instances (Ubuntu Server 18.04 LTS, 1 Core, 3 GB RAM, and no replicas) [14]. Likewise, six SQL instances of a MariaDB Galera cluster and three NoSQL instances were used. The second AKS group used a traditional SQL instance of MariaDB, with the aim of identifying an operational profile needed to host the microservices on SPIDEP in order to maximise the overall performance of the application and its analysis. In addition, we used Apache as an endpoint and NGINX as a load balancer for the implementation. Various user requests are first received by NGINX, and it sends these requests to the corresponding microservices to append to the requested URI.

To carry out the tests, scripts with random variables developed in Apache JMeter (5.2.1) were used to measure the load of virtual users and the performance behaviour of each microservice, together with BlazeMeter servers (US East [Virginia, Google] and EU West [Frankfurt, Google]) in order to generate the workload of the microservices [14], [19], [59].

The default scenario involved accessing the user login (without administrator privileges) and randomly executing a specific instruction for each service (AH), except for Service I (ninth), since this is at an experimental stage of integration with the CDSS [6].

It should be noted that SPIDEP has a medical data set of 6,920 records, and the objective was to generate a workload that drastically increased (10 in 10 active users every five minutes) to emulate a specific number of concurrent user sessions (50 virtual users) according to the specifications of the container (one core, 3 GB RAM).

As a follow-up to this activity, two AKS environments for implementing microservices were used to perform the tests and compare the results. The first environment was based on the implementation of new services with hybrid instances in MariaDB Galera Cluster and NoSQL (SPIDEPMS- T1-HB), while the second was based on the implementation of nine services with a traditional SQL instance of MariaDB (SPIDEPMS-T2-SQL).

We ran the load tests for 20 minutes, with the first test of 50 virtual users on a server located in the USA (NA-Virginia), and the second on a server located in Europe (EU-Frankfurt), since different nursing homes are located on these continents.

To evaluate the performance of the SPIDEP microservices, the results were captured using tabular output reports that were generated after the load tests; these are shown in Tables 1 and 2, which are divided into nine labels (all of the microservices executed [A-H & ALL]).

TABLE 1. Load testing report for SPIDEPMS-T1-HB (time measurements are averages in ms). Each report contains several important values: number of samples, average response time, number of requests processed (hits per second), 90th percentile, 95th percentile, 99th percentile, number and percentage of failed requests, average latency time, and server region.

Label	Samples	Avg. response time	Avg. hits/s	90% line	95% line	99% line	Error count and percentage	Avg. latency	Server location
S-A	2519	1719.470	2.099	2911.000	3119.000	4255.000	0 (0%)	344.504	EU (Germany)
S-B	2512	2112.240	2.099	3615.000	3919.000	4415.000	0 (0%)	358.094	EU (Germany)
S-C	2503	1571.982	2.091	2687.000	2895.000	3199.000	0 (0%)	1548.311	EU (Germany)
S-D	2501	1551.390	2.089	2623.000	2831.000	3119.000	0 (0%)	1529.879	EU (Germany)
S-E	2497	1600.614	2.086	2703.000	2911.000	3199.000	0 (0%)	1577.792	EU (Germany)
S-F	2492	2094.693	2.084	3583.000	3823.000	4191.000	1 (0.04%)	350.259	EU (Germany)
S-G	2483	2105.772	2.076	3615.000	3839.000	4223.000	0 (0%)	340.979	EU (Germany)
S-H	2478	1678.029	2.074	2879.000	3103.000	3487.000	0 (0%)	1654.399	EU (Germany)
ALL	*19985*	*1804.073*	*16.654*	*3119.000*	*3503.000*	*4095.000*	*1 (0.05%)*	*962.057*	*EU (Germany)*
S-A	2404	1854.294	2.007	2895.000	3119.000	4511.000	0 (0%)	479.283	NA (United States)
S-B	2402	2180.154	2.007	3583.000	3807.000	4255.000	0 (0%)	415.681	NA (United States)
S-C	2397	1671.887	2.003	2703.000	2863.000	3215.000	0 (0%)	1586.900	NA (United States)
S-D	2392	1630.239	2.000	2623.000	2815.000	3183.000	0 (0%)	1545.402	NA (United States)
S-E	2386	1669.814	1.997	2703.000	2879.000	3215.000	0 (0%)	1584.658	NA (United States)
S-F	2383	2167.050	1.994	3519.000	3791.000	4127.000	0 (0%)	428.285	NA (United States)
S-G	2371	2186.528	1.986	3519.000	3775.000	4191.000	0 (0%)	443.802	NA (United States)
S-H	2362	1743.218	1.980	2831.000	3039.000	3423.000	0 (0%)	1658.582	NA (United States)
ALL	*19097*	*1887.766*	*15.914*	*3103.000*	*3439.000*	*4063.000*	*0 (0%)*	*1016.982*	*NA (United States)*

TABLE 2. Load testing report for SPIDEPMS-T2-SQL (time measurements are averages in ms). Each report contains several important values: number of samples, average response time, number of requests processed (hits per second), 90th percentile, 95th percentile, 99th percentile, number and percentage of failed requests, average latency time, and server region.

Label	Samples	Avg. response time	Avg. hits/s	90% line	95% line	99% line	Error count and percentage	Avg. latency	Server location
S-A	7676	527.286	6.397	639.000	795.000	2143.000	7636 (99.5%)	464.612	EU (Germany)
S-B	7671	489.840	6.446	615.000	723.000	2175.000	7630 (99.5%)	418.627	EU (Germany)
S-C	7665	673.494	6.512	611.000	719.000	10495.000	7445 (97.1%)	673.138	EU (Germany)
S-D	7657	709.801	6.567	611.000	747.000	10367.000	7403 (96.7%)	709.383	EU (Germany)
S-E	7648	680.486	6.622	615.000	715.000	10559.000	7425 (97.1%)	679.954	EU (Germany)
S-F	7642	489.999	6.680	607.000	711.000	2159.000	7605 (99.5%)	421.735	EU (Germany)
S-G	7635	473.286	6.757	607.000	719.000	2111.000	7605 (99.6%)	413.221	EU (Germany)
S-H	7632	669.365	6.826	611.000	711.000	11071.000	7431 (97.4%)	668.887	EU (Germany)
ALL	*61226*	*589.164*	*51.022*	*615.000*	*727.000*	*10431.000*	*60180 (98.3%)*	*556.149*	*EU (Germany)*
S-A	6131	644.917	5.109	571.000	915.000	4191.000	6082 (99.2%)	559.790	NA (United States)
S-B	6125	616.532	5.147	543.000	715.000	2159.000	6083 (99.3%)	525.174	NA (United States)
S-C	6122	891.623	5.201	547.000	715.000	10623.000	5876 (96.0%)	888.171	NA (United States)
S-D	6111	802.986	5.241	535.000	675.000	10431.000	5915 (96.8%)	800.232	NA (United States)
S-E	6106	861.751	5.282	539.000	703.000	10687.000	5886 (96.4%)	857.184	NA (United States)
S-F	6096	629.875	5.324	547.000	735.000	5215.000	6052 (99.3%)	533.966	NA (United States)
S-G	6092	606.190	5.377	543.000	699.000	2207.000	6055 (99.4%)	524.718	NA (United States)
S-H	6090	846.392	5.438	543.000	683.000	11327.000	5885 (96.6%)	842.092	NA (United States)
ALL	*48873*	*737.500*	*40.728*	*543.000*	*711.000*	*10687.000*	*47834 (97.9%)*	*691.374*	*NA (United States)*

To determine the versatility and robust architecture we have analyzed the global behaviors of the microservices in a real environment of constant and gradual stress, as shown in Figs. 3 and 4. The first scenario (SPIDEPMS-T1-HB) showed that the average response time and the performance statistics (request/hits per second, 90th percentile and 95th percentile), correspond to the established acceptance criteria (5,000 ms); that is, keeping up with the behavior of the identified parameters we can appreciate that the trend increases in a constant and controlled way, according to the amount of active users per second consulting simultaneously the microservices from different regions (United States and Germany) according to the specifications of the container (one core, 3 GB RAM). Meanwhile, the second scenario (SPIDEPMS-T2SQL) showed a chaotic and irregular behavior up to the point of total collapse of SPIDEP Platform.

Several stress simulations were also carried out on the infrastructure hosting SPIDEP, with the aim of exposing the problems that may arise on the platform. Regarding CPU performance, memory, network I/O and connections (Engine Health - BlazeMeter) [70], we analyzed whether the SPIDEP infrastructure itself is capable of supporting the bottlenecks

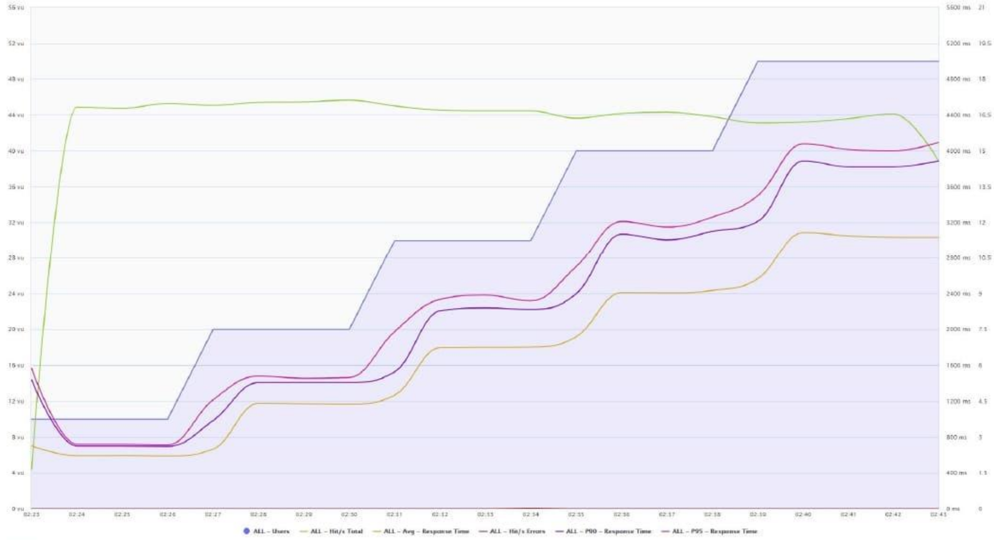

FIGURE 3. Microservice performance patterns for SPIDEPMS-T1-HB (first scenario). The coloured lines represent various parameters: blue: active connections (vu); light green: request/hits per second; red: error/hits per second; light orange: average response time; magenta: 90th percentile response time; purple: 95th percentile response time.

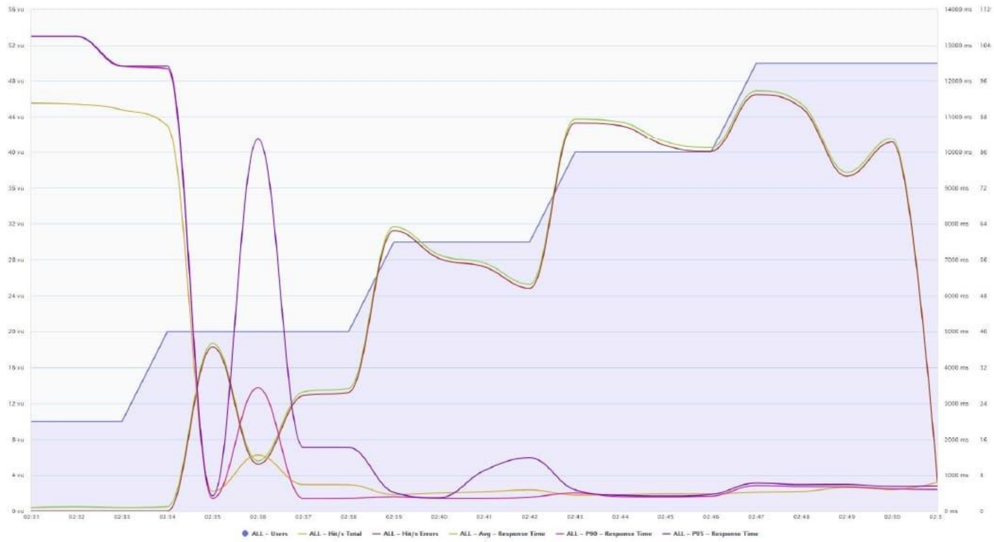

FIGURE 4. Microservice performance patterns for SPIDEPMS-T2-SQL (second scenario). The coloured lines represent various parameters: blue: active connections (vu); light green: request/hits per second; red: error/hits per second; light orange: average response time; magenta: 90th percentile response time; purple: 95th percentile response time.

or errors that occur due to demand; however, the only component that differs in each application is the bandwidth traffic (network I/O), as shown in Figs. 5 and 6.

V. DISCUSSION

To carry out the analysis of the data obtained from the previous section (validation of results), we based ourselves on the

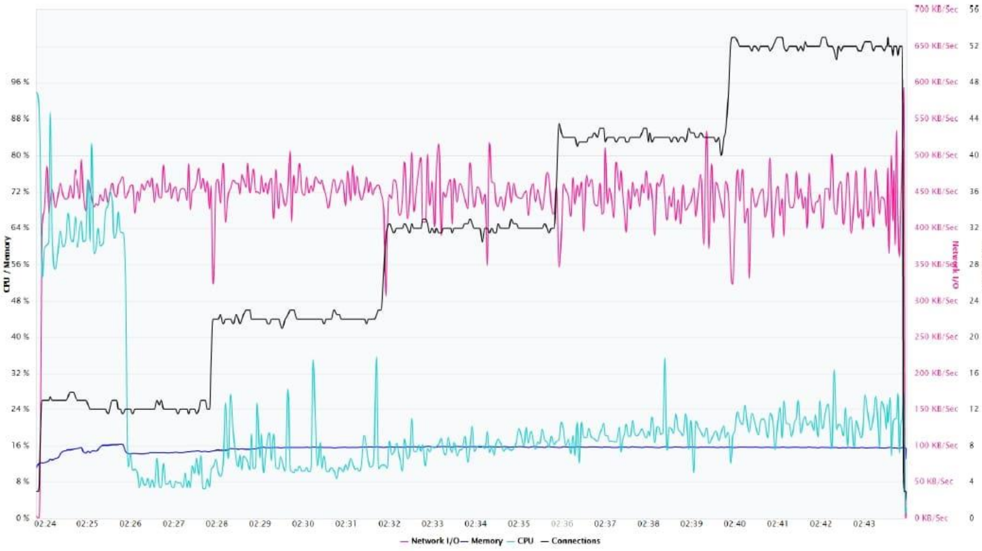

FIGURE 5. Engine Health Report for SPIDEPMS-T1-HB. The coloured lines represent various parameters: purple: bandwidth traffic; blue: memory load generated by users; light blue: CPU load generated by users; black: active connections within the test.

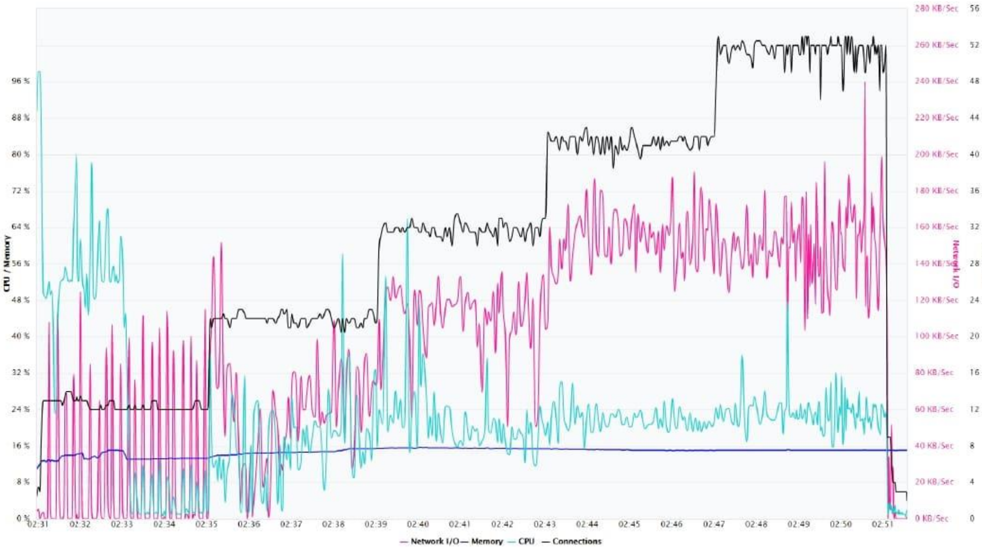

FIGURE 6. Engine Health Report for SPIDEPMS-T2-SQL. The coloured lines represent various parameters: purple: bandwidth traffic; blue: memory load generated by users; light blue: CPU load generated by users; black: active connections within the test.

case studies of the Francisco de Vitoria and Cisneros nursing home in Spain and Carls George and Ntra. Sra del Carmen medical elderly care institution at Dominican Republic, who generously collaborated with their work to obtain the clinical and physiological data of the residents (6,920 records for this study) [6], [9].

This study was carried out with the objective of evaluating the performance and the analysis of each microservice applied to the SPIDEP project within the context of e-health, and for which we analysed, identified and described the main tasks adopted for the design and use of the proposals to perform this work [14].

There are several significant findings from this study. Firstly, it is important to consider acceptance criteria such as the average response time of requests, 90th percentile and their relationship to the percentage of failed requests. The following criteria were established based on various research studies [19], [69], [71]. Criterion A: an acceptable limit for the average response times of the queries and the 90th percentile was 5,000 ms (5 s); criterion B: the acceptable limit for the percentage of failed requests was 5%. Based on these established limits for each test output, any value exceeding these limits was considered to show unacceptable performance.

In view of the above, it can be deduced from Table 1 (SPIDEPMS-T1-HB) that the tests with 50 users were acceptable, since they met the two defined criteria. However, it can be observed from the results in Table 2 (SPIDEPMS-T2-SQL) that the performance is not acceptable: although the results obtained from tests (EU and NA) under criterion A met the established limits, criterion B (percentage of failed requests) was not met, as it exceeded 5%. Since one of the two criteria was not met, the scenario is considered unacceptable.

Regarding the second finding, it is important to note that we observed that microservices created without taking into account the weaknesses of this technology mentioned above, for instance without considering the effects on performance of breaking down an application into multiple services, not contemplate the correct data segmentation (single instance) or not enable a cloud infrastructure for the operation of microservices [19], [72], [73]. Due to the increase in network traffic between microservices and HTTP resource APIs, there appears to be an inherent negative correlation between increased latency and an overall degradation in the performance of the application until collapse is seen.

Additionally, in the third finding, confirmed that the manually created microservices (unsupervised) based on single traditional SQL or software engineering principles do not always yield better performance compared to the monolithic implementation [19], as shown in Fig. 4. In conclusion, to guarantee the versatility and robustness of the architecture, three aspects are important to take into account: (i) scalability, microservices can be scaled individually when running a heavy workload just by replicating them on several containers without and not replicating the others underutilized (maximize the performance with minimal cost) [19], [74]; (ii) communication in a microservice, is important establishing simple communication protocol like http, http-rest "request/response" (synchronous protocol) or mqtt "publish/subscribe" (asynchronous protocol), depending on needs [67]; (iii) fine-grained microservices, it is fundamental to decompose each service focused on a specific function and of limited influence, according to the established requirements [75]; otherwise, this architecture suffers from a high level of abstraction and coordination among the teams [67], [76].

In terms of the fourth finding, both tests (SPIDEPMS-T1-HB and SPIDEPMS-T2-SQL) met the performance criteria for infrastructure (CPU and RAM) under demand from 50 users, since the CPU values were lower than 80% (15% $\sim$ 35%) and memory levels were less than 70% (10% $\sim$ 20%), as shown in Figs. 5 and 6. However, the network I/O value for SPIDEPMS-T2-SQL showed that there was saturation in the bandwidth traffic between the services and the traditional SQL instance (a single database for all services). Consequently, there was a high possibility of packet loss between services and data (communication). Due to this last factor, a high percentage of failed requests occurred, as shown in Fig. 6.

VI. CONCLUSIONS AND FUTURE WORK

This work has analysed, described and justified all the steps involved in the design, development and implementation of the e-health services that make up the SPIDEP project platform. This platform is based on a five-tier architecture using microservices. The objective of the platform is to create a framework based on the use of new ICT to support the early diagnosis of infectious diseases with the added benefits of increasing the level of service in medical care and reductions in cost.

The proposed e-health system is also suitable for distributed computing and for the use of big data and NoSQL structures that allow for the immediate application of machine learning and AI algorithms to discover hidden patterns in the health data of this population. Above all, however, the key innovation that brings telemonitoring system is the possibility of obtaining the medical information necessary to build analytical and predictive e-health models.

In addition, this proposal paves the way for future research for the design of services, based on the proposed workflow for automation, integration and continuous deployment of services, with the aim of achieving greater performance in terms of organisation and patient care.

The work done here is currently being expanded by exploring the possibility of further optimising the performance of SPIDEP by implementing microservices in other programming language environments (paradigms), or by integrating Kafka or another platform for event-driven management within Kubernetes, in order to allow for comparisons between versions. In addition, integration tests (SPIDEP-CDSS) have been planned with various practical studies (real data from a patient group) of the ninth service (Service I) since this work is currently at the experimental stage.

Another possible area of future research would be to incorporate the perspective of data mining by developing microservices adapted to machine learning (e.g., algorithms for early prediction of COVID-19 [77]), with the aim of identifying patterns in COVID-19 behavior; however, it would be necessary to provide sufficient time-space series data to begin training the required algorithms [78]. For this reason, the research group is currently implementing a national geolocalized repository containing health and referral data of patients infected during the COVID-19 epidemic in Panama (proposed case study), funded by the National Secretariat of Science, Technology and Innovation of Panama (SENACYT-PANAMA).

In this sense, these data collected by the regional repository will provide various demographic variables that affect the rate of spread, such as: blocking variables, variability in infected populations, socioeconomic information and others [79], [80]. The georeferenced results of these analyses represent novel microservices for evaluating the development of the disease over time, considering demographic data and implementing effective measures to contain the spread of future epidemiological events such as social distancing and health fences, among others [80].

Further researches are needed to obtain sufficient results that can demonstrate the robustness of the proposed architecture in terms of the adaptations of the microservices to identify patterns of COVID-19 behavior or other epidemiological events.

ACKNOWLEDGEMENTS

The authors would like to extend special thanks to all the clinical and healthcare staff of the nursing homes for the elderly, the Francisco de Vitoria and Cisneros centres of the Department of Social Policies and Family of the Community of Madrid in Spain, and Ntra. Sra del Carmen and Carls George in the Dominican Republic, who generously worked with us to obtain the medical data of the residents.

REFERENCES

[1] D. Mendes, D. Jorge, G. Pires, R. Panda, R. António, P. Dias, and L. Oliveira, "VITASENIOR-MT: A distributed and scalable cloud-based telehealth solution," in *Proc. IEEE 5th World Forum Internet Things (WF-IoT)*, Limerick, Ireland, Apr. 2019, pp. 767–772, doi: 10.1109/WF-IoT.2019.8767184.

[2] V. Kontis, J. E. Bennett, C. D. Mathers, G. Li, K. Foreman, and M. Ezzati, "Future life expectancy in 35 industrialised countries: Projections with a Bayesian model ensemble," *Lancet*, vol. 389, no. 10076, pp. 1323–1335, Apr. 2017, doi: 10.1016/S0140-6736(16)32381-9.

[3] M. Solé-Casals, E. Chirveches-Pérez, E. Puigoriol-Juvanteny, N. Nubó-Puntí, C. Chabrera-Sanz, and M. Subirana-Casacuberta, "Perfil y resultados del paciente frágil valorado por la enfermera de Práctica avanzada en un servicio de urgencias," *Enfermería Clínica*, vol. 28, no. 6, pp. 365–374, Nov. 2018, doi: 10.1016/J.ENFCLI.2017.04.003.

[4] A. Baldominos, A. Puello, H. Ogul, T. Asuroglu, and R. Colomo-Palacios, "Predicting infections using computational intelligence—A systematic review," *IEEE Access*, vol. 8, pp. 31083–31102, 2020, doi: 10.1109/ACCESS.2020.2973006.

[5] J. Sanz-Moreno, J. Gómez-Pulido, A. Garcés, H. Calderón-Gómez, M. Vargas-Lombardo, J. L. Castillo-Sequera, M. L. P. Luque, R. Toro, and G. Sención-Martínez, "mHealth system for the early detection of infectious diseases using biomedical signals," in *Advances in Automation and Robotics Research*. Cham, Switzerland: Springer, 2020, pp. 203–213, doi: 10.1007/978-3-030-40309-6_20.

[6] A. Baldominos, H. Ogul, R. Colomo-Palacios, J. Sanz-Moreno, and J. M. Gómez-Pulido, "Infection prediction using physiological and social data in social environments," *Inf. Process. Manage.*, vol. 57, no. 3, May 2020, Art. no. 102213, doi: 10.1016/J.IPM.2020.102213.

[7] F. Raschilas, H. Blain, and C. Jeandel, "La infección en el paciente de edad avanzada," *EMC-Tratado Med.*, vol. 10, no. 2, pp. 1–11, Jan. 2006, doi: 10.1016/S1636-5410(06)70381-6.

[8] E. Á. Artero, A. C. Nuñez, M. G. Bravo, O. C. Calvo, M. B. Garcia, and J. P. Lledias, "Infección urinaria en el anciano," *Revista Clínica Española*, vol. 219, no. 4, pp. 189–193, May 2019, doi: 10.1016/J.RCE.2018.10.009.

[9] H. Calderón-Gómez, A. Garcés-Jiménez, M. Vargas-Lombardo, J. M. Gómez-Pulido, M.-L. Polo-Luque, J. L. Castillo, G. Sención, and J. S. Moreno, "Proposal using the cloud architecture in system for the early detection of infectious diseases in elderly people fed by biosensors records," in *Proc. 7th Int. Eng., Sci. Technol. Conf. (IESTEC)*, Panama City, Panama, 2019, pp. 631–634, doi: 10.1109/IESTEC46403.2019.00118.

[10] M. A. Martín Martínez, R. C. Alférez, E. Escortell Mayor, M. Rico Blázquez, and A. Sarría Santamera, "Factores asociados a reingresos hospitalarios en pacientes de edad avanzada," *Atención Primaria*, vol. 43, no. 3, pp. 117–124, Mar. 2011, doi: 10.1016/J.APRIM.2009.12.007.

[11] S. M. Q. Vera, D. M. Rojas Aguilar, D. A. Chavarro-Carvajal, and I. Riaño Forero, "Mortalidad en pacientes mayores de 65 años ingresados en cuidados intensivos del hospital universitario san ignacio en el 2014," *Acta Colombiana de Cuidado Intensivo*, vol. 19, no. 2, pp. 61–68, Apr. 2019, doi: 10.1016/J.ACCI.2018.11.002.

[12] B. Dinesen, B. Nonnecke, D. Lindeman, E. Toft, K. Kidholm, K. Jethwani, H. M. Young, H. Spindler, C. U. Oestergaard, J. A. Southard, M. Gutierrez, N. Anderson, N. M. Albert, J. J. Han, and T. Nesbitt, "Personalized telehealth in the future: A global research agenda," *J. Med. Internet Res.*, vol. 18, no. 3, p. e53, Mar. 2016, doi: 10.2196/jmir.5257.

[13] EU-CELAC. Fundación para la Investigación Biomédica del Hospital Universitario Príncipe de Asturias, (2017). *Design and Implementation of a Low Cost Smart System for Pre-Diagnosis and Telecare of Infectious Diseases in Elderly People | EU-CELAC*. Accessed: Feb. 11, 2020. [Online]. Available: https://www.eucelac-platform.eu/project/design-and-implementation-low-cost-smart-system-pre-diagnosis-and-telecare-infectious

[14] H. Calderón-Gómez *et al.*, "Development of ehealth applications-based on microservices in a cloud architecture," *RISTI Rev. Iber. Sist. Tecnol. Inf.*, vol. 2019, no. E23, pp. 81–93, Oct. 2019.

[15] D. C. Robinson, S. Mohanty, J. Young, G. Jones, and D. Wesemann, "Novel techniques for mapping infectious diseases using point of care diagnostic sensors," in *Proc. 2nd Int. Symp. Phys. Technol. Sensors (ISPTS)*, Pune, India, 2015, pp. 14–18, doi: 10.1109/ISPTS.2015.7220148.

[16] D. Guaman, L. Yaguachi, C. C. Samanta, J. H. Danilo, and F. Soto, "Performance evaluation in the migration process from a monolithic application to microservices," in *Proc. 13th Iberian Conf. Inf. Syst. Technol. (CISTI)*, Caceres, Jun. 2018, pp. 1–8, doi: 10.23919/CISTI.2018.8399148.

[17] S. Newman, *Building Microservices: Designing Fine-Grained Systems*, 1st ed. Newton, MA, USA: O'Reilly Media, 2015, p. 280.

[18] H. Jita and V. Pieterse, "A framework to apply the Internet of Things for medical care in a home environment," in *Proc. Int. Conf. Cloud Comput. Internet Things*, 2018, pp. 45–54, doi: 10.1145/3291064.3291065.

[19] M. Abdullah, W. Iqbal, and A. Erradi, "Unsupervised learning approach for Web application auto-decomposition into microservices," *J. Syst. Softw.*, vol. 151, pp. 243–257, May 2019, doi: 10.1016/J.JSS.2019.02.031.

[20] D. Namiot and M. Sneps-Sneppe, "On micro-services architecture," *Int. J. Open Inf. Technol.*, vol. 2, no. 9, pp. 24–27, 2014.

[21] P. Di Francesco, P. Lago, and I. Malavolta, "Architecting with microservices: A systematic mapping study," *J. Syst. Softw.*, vol. 150, pp. 77–97, Apr. 2019, doi: 10.1016/J.JSS.2019.01.001.

[22] I. Odun-Ayo, R. Goddy-Worlu, V. Geteloma, and E. Grant, "A systematic mapping study of cloud, fog, and edge/mobile devices management, hierarchy models and business models," *Adv. Sci., Technol. Eng. Syst. J.*, vol. 4, no. 2, pp. 91–101, 2019, doi: 10.25046/aj040212.

[23] A. Al-Qamash, I. Soliman, R. Abulibdeh, and M. Saleh, "Cloud, fog, and edge computing: A software engineering perspective," in *Proc. Int. Conf. Comput. Appl. (ICCA)*, Aug. 2018, pp. 276–284, doi: 10.1109/COMAPP.2018.8460443.

[24] A. Grguriš, M. Mošmondor, and D. Huljenić, "The SmartHabits: An intelligent privacy-aware home care assistance system," *Sensors*, vol. 19, no. 4, p. 907, Feb. 2019, doi: 10.3390/s19040907.

[25] S. Roca, J. Sancho, J. García, and Á. Alesanco, "Microservice chatbot architecture for chronic patient support," *J. Biomed. Informat.*, vol. 102, Feb. 2020, Art. no. 103305, doi: 10.1016/J.JBI.2019.103305.

[26] I. Semenov, R. Osenev, S. Gerasimov, G. Kopanitsa, D. Denisov, and Y. Andreychuk, "Experience in developing an FHIR medical data management platform to provide clinical decision support," *Int. J. Environ. Res. Public Health*, vol. 17, no. 1, p. 73, Dec. 2019, doi: 10.3390/ijerph17010073.

[27] J. M. Alvarez, J. A. Sanabria, and J. I. Garcia, "Microservices-based architecture for fault diagnosis in tele-rehabilitation equipment operated via Internet," in *Proc. IEEE Latin Amer. Test Symp. (LATS)*, Santiago, Chile, Mar. 2019, pp. 1–6, doi: 10.1109/LATW.2019.8704556.

[28] C. Andrikos, G. Rassias, P. Tsanakas, and I. Maglogiannis, "An enhanced device-transparent real-time teleconsultation environment for radiologists," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 1, pp. 374–386, Jan. 2019, doi: 10.1109/JBHI.2018.2824312.

[29] P. E. Khoonsari *et al.*, "Interoperable and scalable data analysis with microservices: Applications in metabolomics," *Bioinformatics*, vol. 35, no. 19, pp. 3752–3760, Mar. 2019, doi: 10.1093/bioinformatics/btz160.

[30] M. Saadeh, A. Sleit, K. E. Sabri, and W. Almobaideen, "Hierarchical architecture and protocol for mobile object authentication in the context of IoT smart cities," *J. Netw. Comput. Appl.*, vol. 121, pp. 1–19, Nov. 2018, doi: 10.1016/J.JNCA.2018.07.009.

[31] L. Catarinucci, D. de Donno, L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and L. Tarricone, "An IoT-aware architecture for smart healthcare systems," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 515–526, Dec. 2015, doi: 10.1109/jiot.2015.2417684.

[32] A. Javed, J. Robert, K. Heljanko, and K. Främling, "IoTEF: A federated edge-cloud architecture for fault-tolerant IoT applications," *J. Grid Comput.*, vol. 18, no. 1, pp. 57–80, Mar. 2020, doi: 10.1007/s10723-019-09498-8.

[33] A. Sharma, M. Kumar, and S. Agarwal, "A complete survey on software architectural styles and patterns," *Procedia Comput. Sci.*, vol. 70, pp. 16–28, 2015, doi: 10.1016/j.procs.2015.10.019.

[34] D. Glushkova, P. Jovanovic, and A. Abelló, "Mapreduce performance model for Hadoop 2.X," *Inf. Syst.*, vol. 79, pp. 32–43, Jan. 2019, doi: 10.1016/J.IS.2017.11.006.

[35] R. Hernandez, Y. Becerra, J. Torres, and E. Ayguade, "Automatic query driven data modelling in cassandra," *Procedia Comput. Sci.*, vol. 51, pp. 2822–2826, Jan. 2015, doi: 10.1016/J.PROCS.2015.05.441.

[36] O. Bello and S. Zeadally, "Intelligent device-to-device communication in the Internet of Things," *IEEE Syst. J.*, vol. 10, no. 3, pp. 1172–1182, Sep. 2016, doi: 10.1109/JSYST.2014.2298837.

[37] X. Wan, X. Guan, T. Wang, G. Bai, and B.-Y. Choi, "Application deployment using microservice and docker containers: Framework and optimization," *J. Netw. Comput. Appl.*, vol. 119, pp. 97–109, Oct. 2018, doi: 10.1016/j.jnca.2018.07.003.

[38] S. Taherizadeh and M. Grobelnik, "Key influencing factors of the kubernetes auto-scaler for computing-intensive microservice-native cloud-based applications," *Adv. Eng. Softw.*, vol. 140, Feb. 2020, Art. no. 102734, doi: 10.1016/J.ADVENGSOFT.2019.102734.

[39] T. Kiss, P. Kacsuk, J. Kovacs, B. Rakoczi, A. Hajnal, A. Farkas, G. Gesmier, and G. Terstyanszky, "MiCADO—Microservice-based cloud application-level dynamic orchestrator," *Future Gener. Comput. Syst.*, vol. 94, pp. 937–946, May 2019, doi: 10.1016/J.FUTURE.2017.09.050.

[40] B. Götz, D. Schel, D. Bauer, C. Henkel, P. Einberger, and T. Bauernhansl, "Challenges of production microservices," *Procedia CIRP*, vol. 67, pp. 167–172, 2018, doi: 10.1016/J.PROCIR.2017.12.194.

[41] M. Krämer, S. Frese, and A. Kuijper, "Implementing secure applications in smart city clouds using microservices," *Future Gener. Comput. Syst.*, vol. 99, pp. 308–320, Oct. 2019, doi: 10.1016/J.FUTURE.2019.04.042.

[42] M. Gribaudo, M. Iacono, and D. Manini, "Performance evaluation of replication policies in microservice based architectures," *Electron. Notes Theor. Comput. Sci.*, vol. 337, pp. 45–65, May 2018, doi: 10.1016/J.ENTCS.2018.03.033.

[43] H. Suryotrisongko, D. P. Jayanto, and A. Tjahyanto, "Design and development of backend application for public complaint systems using microservice spring boot," *Procedia Comput. Sci.*, vol. 124, pp. 736–743, Jan. 2017, doi: 10.1016/J.PROCS.2017.12.212.

[44] N. Dragoni, S. Giallorenzo, A. L. Lafuente, M. Mazzara, F. Montesi, R. Mustafin, L. Safina, "Microservices: Yesterday, today, and tomorrow," in *Present and Ulterior Software Engineering*. Cham, Switzerland: Springer, 2017, pp. 195–216.

[45] P. W. Handayani, A. N. Hidayanto, A. A. Pinem, I. C. Hapsari, P. I. Sandhyaduhita, and I. Budi, "Acceptance model of a hospital information system," *Int. J. Med. Informat.*, vol. 99, pp. 11–28, Mar. 2017, doi: 10.1016/J.IJMEDINF.2016.12.004.

[46] C. Catley and M. Frize, "Design of a health care architecture for medical data interoperability and application integration," in *Proc. 2nd Joint 24th Annu. Conf. Annu. Fall Meeting Biomed. Eng. Soc. Eng. Med. Biol.*, vol. 3, 2002, pp. 1952–1953, doi: 10.1109/IEMBS.2002.1053111.

[47] W. Hasselbring, "Microservices for scalability: Keynote talk abstract," in *Proc. 7th ACM/SPEC Int. Conf. Perform. Eng.*, New York, NY, USA, 2016, pp. 133–134, doi: 10.1145/2851553.2858659.

[48] A. R. Sampaio, J. Rubin, I. Beschastnikh, and N. S. Rosa, "Improving microservice-based applications with runtime placement adaptation," *J. Internet Services Appl.*, vol. 10, no. 1, p. 4, Dec. 2019, doi: 10.1186/s13174-019-0104-0.

[49] C. Chen, "With great abstraction comes great responsibility: Sealing the microservices attack surface," in *Proc. IEEE Cybersecur. Develop. (SecDev)*, Tysons Corner, VA, USA, Sep. 2019, p. 144, doi: 10.1109/SecDev.2019.00027.

[50] Y. Yu, J. Yang, C. Guo, H. Zheng, and J. He, "Joint optimization of service request routing and instance placement in the microservice system," *J. Netw. Comput. Appl.*, vol. 147, Dec. 2019, Art. no. 102441, doi: 10.1016/J.JNCA.2019.102441.

[51] R. Heinrich, A. V. Hoorn, H. Knoche, F. Li, L. E. Lwakatare, C. Pahl, S. Schulte, and J. Wettinger, "Performance engineering for microservices: Research challenges and directions," in *Proc. 8th ACM/SPEC Int. Conf. Perform. Eng. Companion*, New York, NY, USA, 2017, pp. 223–226, doi: 10.1145/3053600.3053653.

[52] C. Fan and S. Ma, "Migrating monolithic mobile application to microservice architecture: An experiment report," in *Proc. IEEE Int. Conf. AI Mobile Services (AIMS)*, Honolulu, HI, USA, Jun. 2017, pp. 109–112, doi: 10.1109/AIMS.2017.23.

[53] S.-P. Ma, C.-Y. Fan, Y. Chuang, I.-H. Liu, and C.-W. Lan, "Graph-based and scenario-driven microservice analysis, retrieval, and testing," *Future Gener. Comput. Syst.*, vol. 100, pp. 724–735, Nov. 2019, doi: 10.1016/J.FUTURE.2019.05.048.

[54] K. Jander, L. Braubach, and A. Pokahr, "Defense-in-depth and role authentication for microservice systems," *Procedia Comput. Sci.*, vol. 130, pp. 456–463, Jan. 2018, doi: 10.1016/J.PROCS.2018.04.047.

[55] P. Raković and B. Lutovac, "A cloud computing architecture with wireless body area network for professional athletes health monitoring in sports organizations—Case study of Montenegro," in *Proc. 4th Medit. Conf. Embedded Comput. (MECO)*, Budva, Montenegro, 2015, pp. 387–390, doi: 10.1109/MECO.2015.7181950.

[56] A. Domínguez and M. Vargas-Lombardo, "El estado del arte: Salud inteligente y el Internet de las cosas," *I+D Tecnológico*, vol. 14, no. 1, pp. 14–17, Jun. 2018, doi: 10.33412/idt.v14.1.1809.

[57] A. Fatima and R. Colomo-Palacios, "Security aspects in healthcare information systems: A systematic mapping," *Procedia Comput. Sci.*, vol. 138, pp. 12–19, Jan. 2018, doi: 10.1016/J.PROCS.2018.10.003.

[58] S. Li, H. Zhang, Z. Jia, Z. Li, C. Zhang, J. Li, Q. Gao, J. Ge, and Z. Shan, "A dataflow-driven approach to identifying microservices from monolithic applications," *J. Syst. Softw.*, vol. 157, Nov. 2019, Art. no. 110380, doi: 10.1016/J.JSS.2019.07.008.

[59] Y. Gan and C. Delimitrou, "The architectural implications of cloud microservices," *IEEE Comput. Archit. Lett.*, vol. 17, no. 2, pp. 155–158, Jul. 2018, doi: 10.1109/LCA.2018.2839189.

[60] Á. Brandón, M. Solé, A. Huélamo, D. Solans, M. S. Pérez, and V. Muntés-Mulero, "Graph-based root cause analysis for service-oriented and microservice architectures," *J. Syst. Softw.*, vol. 159, Jan. 2020, Art. no. 110432, doi: 10.1016/J.JSS.2019.110432.

[61] F. Rademacher, S. Sachweh, and A. Zündorf, "Aspect-oriented modeling of technology heterogeneity in microservice architecture," in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Mar. 2019, pp. 21–30, doi: 10.1109/ICSA.2019.00011.

[62] F. H. Vera-Rivera, "Método de automatización del despliegue continuo en la nube para la implementación de microservicios," in *Proc. 21st Conf. Iberoamericana Ingeniería Softw. (CIBSE)*, Bogotá, Colombia, 2018, pp. 597–604.

[63] H. Vural, M. Koyuncu, and S. Guney, "A systematic literature review on microservices BT—Computational science and its applications," in *Proc. ICCSA*, Trieste, Italy, 2017, pp. 203–217.

[64] L. Bao, C. Wu, X. Bu, N. Ren, and M. Shen, "Performance modeling and workflow scheduling of microservice-based applications in clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 2114–2129, Sep. 2019, doi: 10.1109/TPDS.2019.2901467.

[65] S. N. Srirama, M. Adhikari, and S. Paul, "Application deployment using containers with auto-scaling for microservices in cloud environment," *J. Netw. Comput. Appl.*, vol. 160, Jun. 2020, Art. no. 102629, doi: 10.1016/J.JNCA.2020.102629.

[66] D. Morris, S. Voutsinas, N. C. Hambly, and R. G. Mann, "Use of docker for deployment and testing of astronomy software," *Astron. Comput.*, vol. 20, pp. 105–119, Jul. 2017, doi: 10.1016/j.ascom.2017.07.004.

[67] A. Benayache, A. Bilami, S. Barkat, P. Lorenz, and H. Taleb, "MsM: A microservice middleware for smart WSN-based IoT application," *J. Netw. Comput. Appl.*, vol. 144, pp. 138–154, Oct. 2019, doi: 10.1016/j.jnca.2019.06.015.

[68] F. Lonetti and E. Marchetti, "Emerging software testing technologies," *Adv. Comput.*, vol. 108, pp. 91–143, Jan. 2018, doi: 10.1016/BS.ADCOM.2017.11.003.

[69] R. Khan and M. Amjad, "Performance testing (load) of Web applications based on test case management," *Perspect. Sci.*, vol. 8, pp. 355–357, Sep. 2016, doi: 10.1016/J.PISC.2016.04.073.

[70] BlazeMeter. *Engine Health Report*. Accessed: Feb. 7, 2020. [Online]. Available: https://guide.blazemeter.com/hc/en-us/articles/206733869-Engine-Health-Report-Engine-Health-Report

[71] R. Ramakrishnan and A. Kaur, "Little's law based validation framework for load testing," *Inf. Softw. Technol.*, vol. 108, pp. 88–98, Apr. 2019, doi: 10.1016/J.INFSOF.2018.11.007.

[72] F. Alharbi, A. Atkins, C. Stanier, and H. A. Al-Buti, "Strategic value of cloud computing in healthcare organisations using the balanced scorecard approach: A case study from a Saudi hospital," *Procedia Comput. Sci.*, vol. 98, pp. 332–339, Jan. 2016, doi: 10.1016/J.PROCS.2016.09.050.

[73] N. M. Kumar and P. K. Mallick, "The Internet of Things: Insights into the building blocks, component interactions, and architecture layers," *Procedia Comput. Sci.*, vol. 132, pp. 109–117, Jan. 2018, doi: 10.1016/J.PROCS.2018.05.170.

[74] T. Kiss, J. DesLauriers, G. Gesmier, G. Terstyanszky, G. Pierantoni, O. A. Oun, S. J. E. Taylor, A. Anagnostou, and J. Kovacs, "A cloud-agnostic queuing system to support the implementation of deadline-based application execution policies," *Future Gener. Comput. Syst.*, vol. 101, pp. 99–111, Dec. 2019, doi: 10.1016/j.future.2019.05.062.

[75] Z. Yi, W. Meilin, C. RenYuan, W. YangShuai, and W. Jiao, "Research on application of SME manufacturing cloud platform based on micro service architecture," *Procedia CIRP*, vol. 83, pp. 596–600, Jan. 2019, doi: 10.1016/j.procir.2019.04.091.

[76] K. Malyuga, O. Perl, A. Slapoguzov, and I. Perl, "Fault tolerant central saga orchestrator in RESTful architecture," in *Proc. 26th Conf. Open Innov. Assoc. (FRUCT)*, Yaroslavl, Russia, Apr. 2020, pp. 278–283, doi: 10.23919/FRUCT48808.2020.9087389.

[77] L. Zhong, L. Mu, J. Li, J. Wang, Z. Yin, and D. Liu, "Early prediction of the 2019 novel coronavirus outbreak in the mainland China based on simple mathematical model," *IEEE Access*, vol. 8, pp. 51761–51769, 2020, doi: 10.1109/ACCESS.2020.2979599.

[78] S. J. Fong, G. Li, N. Dey, R. G. Crespo, and E. Herrera-Viedma, "Composite Monte Carlo decision making under high uncertainty of novel coronavirus epidemic using hybridized deep learning and fuzzy rule induction," *Appl. Soft Comput.*, vol. 93, Aug. 2020, Art. no. 106282, doi: 10.1016/j.asoc.2020.106282.

[79] D. Dong, Z. Tang, S. Wang, H. Hui, L. Gong, Y. Lu, Z. Xue, H. Liao, F. Chen, F. Yang, R. Jin, K. Wang, Z. Liu, J. Wei, W. Mu, H. Zhang, J. Jiang, J. Tian, and H. Li, "The role of imaging in the detection and management of COVID-19: A review," *IEEE Rev. Biomed. Eng.*, early access, Apr. 27, 2020, doi: 10.1109/RBME.2020.2990959.

[80] L. Li, Q. Zhang, X. Wang, J. Zhang, T. Wang, T.-L. Gao, W. Duan, K. K.-F. Tsoi, and F.-Y. Wang, "Characterizing the propagation of situational information in social media during COVID-19 epidemic: A case study on Weibo," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 2, pp. 556–562, Apr. 2020, doi: 10.1109/TCSS.2020.2980007.

HURIVIADES CALDERÓN-GÓMEZ was born in Panama, in 1993. He received the B.E. degree in software development from the Technological University of Panama, Coclé, Panama, in 2016, and the M.E. degree in higher education from University Santander, Panama City, Panama, in 2019. He is currently pursuing the Ph.D. degree in information and knowledge engineering with the University of Alcalá, Alcalá de Henares, Spain.

From 2014 to 2016, he was a Research Assistant with the Technological University of Panama. In 2017, he joined the Technological University of Panama as a Research Associate. Since March 2019, he has been with the Department of Computer Systems Engineering, Technological University of Panama, where he was an Associate Professor. His current research interests include e-health, cloud computing, and software engineering.

LUIS MENDOZA-PITTÍ was born in Panama, in 1992. He received the B.E. degree in engineering systems and computing from the Technological University of Panama, Panama, in 2015. He is currently pursuing the Ph.D. degree in information and knowledge engineering with the University of Alcalá, Alcalá de Henares, Spain.

From 2013 to 2016, he was a Student Research Assistant with the Technological University of Panama. His current research interests include smart building, energy optimization, cloud computing, and software engineering.

MIGUEL VARGAS-LOMBARDO (Member, IEEE) received the Ph.D. degree from the Polytechnic University of Madrid. He is currently a member of the National Research System (SNI-SENACYT) of Panama. He is currently a Professor and a Researcher with the Technological University of Panama.

JOSÉ MANUEL GÓMEZ-PULIDO was born in Cáceres, Spain. He received the B.S. and M.S. degrees in telecommunications engineering from the University Polytechnic of Madrid, Spain, in 1988 and 1994, respectively, and the Ph.D. degree in telecommunication from the University of Alcalá, Madrid, in 2004. In 1992, he was an Assistant Researcher and a Professor with the University Polytechnic of Madrid. From 1993 to 2004, he was with the Signal Theory and Communications Department, University of Alcalá, where he is currently an Assistant Professor with the Computational Science Department. He has participated in more than 24 research projects several research projects with Spanish and European companies, related with radio propagation in mobile communications, signal processing in radar, ultrasound and laser applications, EMI–EMC design in on-board satellite instrumentation, radio propagation materials proprieties, numerical technics applied in telecommunication, and so on. His research interests include the areas of wireless sensors networks, smart grids, and new technologies in the data mining and artificial intelligence world applied to the ICT.

JOSÉ LUIS CASTILLO-SEQUERA received the degree in computing from the National University of San Marcos, Peru, the master's degree in computer project management and the master's degree in university teaching in Spain, and the Ph.D. degree in information systems, documentation, and knowledge from the University of Alcalá, Spain. He is currently a Full Professor with the Department of Computer Science, University of Alcalá. He has multiple publications at the level of high-level indexed JCR journals and is the author of books and book chapters related to the field of artificial intelligence. His research interests include information retrieval, knowledge extraction based on evolutionary learning and neural network, data mining and big data, documentation, learning, and teaching innovation with Web 2.0. He also leads projects of technological innovation oriented to the use of ICTs in university research groups. His research interests include business technologies and intelligent technologies for knowledge management with interdisciplinary technologies.

JOSÉ SANZ-MORENO was born in Spain, in 1956. He received the degree in internal medicine. He is currently the Chief of the Division of Infectious Diseases, Hospital Universitario Principe de Asturias, Alcalá de Henares, Madrid, Spain. He is also a Professor with the Department of Medicine and Medical Specialties, University of Alcalá, Spain.

GLORIA SENCIÓN received the degree in nephrologist from the Hospital Príncipe de Asturias, Alcalá de Henares, Spain, in 2015. She is currently a Professor with the Department of Medicine, Autonomous University of Santo Domingo, Dominican Republic.

2.3 Artículo 3: "Evaluating Service-Oriented and Microservice Architecture Patterns to Deploy eHealth Applications in Cloud Computing Environment"

2.3.1 Resumen del artículo

En este artículo propone un nuevo marco de trabajo para la concepción de una plataforma eHealth centrada en los entornos de computación en la nube, ya que los enfoques actuales y emergentes con respecto al desarrollo de sistemas de recomendación basados en la telemonitorización y el acceso a la historia digital clínica demuestran cada vez más su potencial para mantener la calidad de las prestaciones en los servicios de atención médica, principalmente en el diagnóstico clínico asistido dentro del campo de las enfermedades infecciosas y el empeoramiento en patologías crónicas. Teniendo en cuenta lo anterior, nuestro objetivo es evaluar y contrastar el rendimiento de los diferentes patrones arquitectónicos más utilizados para la creación de aplicaciones eHealth (Service-Oriented Architectures "SOA" o Microservices Architecture "MSA"), tomando como referencia los valores cuantitativos obtenidos de las diversas pruebas de rendimiento y a su capacidad de adaptarse a las características de softwares requeridas. Por ello, fue necesario la bifurcación de nuestra plataforma, ajustada a dos variantes arquitectónicas (SOA & MSA). Como seguimiento de esta actividad, se realizaron las pruebas correspondientes, las cuales arrojaron que la variante MSA demuestra un mejor funcionamiento en cuanto al rendimiento y al tiempo de respuesta comparado a la variante SOA; sin embargo, el consumo de ancho de banda en MSA fue más significativo que SOA, y la escalabilidad en SOA generalmente no es posible o requiere mucho trabajo para lograrlo. En conclusión, consideramos que la implementación de SOA y MSA depende de la naturaleza y las necesidades de las organizaciones (i.e., rendimiento, disponibilidad o interoperabilidad) [56].

2.3.2 Conclusiones del artículo

En este artículo, se ha descrito la metodología y desarrollos involucrados en el diseño, implementación y despliegue de la plataforma SPIDEP y sus variantes basadas en los patrones arquitectónicos SOA y MSA de una plataforma de eHealth desarrollada y orientada a la telemonitorización remota de personas mayores; por consiguiente, era necesario evaluar cuáles de las variantes de SPIDEP se adaptan mejor a nuestro concepto de versatilidad de alto rendimiento, considerando las fortalezas y debilidades de cada arquitectura y cómo se comportan en un escenario de alta demanda. Como resultado, se ha analizado y contrastado el rendimiento de cada variante frente a las métricas obtenidas en las distintas pruebas de rendimiento (es decir, tiempo de respuesta, uso eficiente de la infraestructura y consumo de red), lo que nos ha permitido constatar que MSA presenta un mejor rendimiento en cuanto al atributo de calidad de rendimiento (aproximadamente un 54,21%). Del mismo modo, al procesar múltiples solicitudes de varios servicios, el tiempo de respuesta fue inferior en comparación con SOA (aproximadamente un 7,34%), pero el consumo de ancho de banda en MSA fue más significativo que en SOA (aproximadamente un 73,80%). Por lo tanto, dentro del contexto de la eHealth y en base a los resultados obtenidos, observamos que MSA es capaz de satisfacer la mayoría de las necesidades de apoyo a la toma de decisiones médicas y es adaptable a diferentes tipos de sistemas clínicos (p. ej. EHR, IoHT o sistemas de telemonitorización) y a diversas soluciones de infraestructura (p. ej. residencias de ancianos, hospitales y gestión de la salud pública); sin embargo, esto conlleva muchos retos y se requiere de un sólido flujograma de para el diseños de cada servicios que conforman el ecosistema. Cabe destacar, que los patrones arquitectónicos SOA y MSA pueden considerarse aliados complementarios para una arquitectura interempresarial que confiera un conjunto de servicios diferentes, en lugar de ser competidores, es decir, combinar los atributos SOA y MSA dentro de un entorno.

2.3.3 Contribuciones del artículo a la comunidad científica

El tercer artículo aporta una rigurosa comparativa entre los diferentes patrones arquitectónicos (Service-Oriented Architectures "SOA" y Microservices Architecture "MSA"), tomando como referencia las fortalezas y las debilidades de cada arquitectura basándose en calidad del servicio y sus correspondientes métricas, según nuestras necesidades en el desarrollo de aplicaciones versátiles de alto rendimiento más acorde al contexto eHealth. Por ello, ha sido necesario ramificar la plataforma SPIDEP en dos variantes, siguiendo con los lineamientos de cada patrón arquitectónico llevada directamente por el tesista tanto en la fase de diseño, implementación y despliegue de la plataforma SPIDEP para analizar cuál de las variantes cumplen óptimamente con las especificaciones planteadas; mientras que, los autores se encargaron en las adaptaciones de los clasificadores para cada variante y en la validación de las inferencias.

A diferencia de los dos artículos anteriores, este artículo se evaluó y contrastó el rendimiento de las variantes SPIDEP mediante valores cuantitativos obtenidos de las diversas pruebas de rendimiento y a su capacidad de adaptarse a las características de softwares. En general, estos resultados indican dos importantes aportaciones: (i) proporcionar una visión general de las diferencias entre SOA y MSA y (ii) identificar las implicaciones del diseño, implementación y despliegue de plataformas orientadas a SOA y MSA [56].

Por lo tanto, con este trabajo se propone un ecosistema intergeneracional para SOA-MSA, con el objetivo de desarrollar una base para la integración e interconexión de las diferentes aplicaciones de eHealth involucradas en las organizaciones médicas en conjunción con microservicios adaptados a sistemas de inferencia basados en inteligencia artificial para realizar tareas especializadas de soporte a la toma de decisiones en la prevención, monitorización y tratamiento de diversas afecciones.

2.3.4 Identificación del artículo

- Título: "Evaluating Service-Oriented and Microservice Architecture Patterns to Deploy eHealth Applications in Cloud Computing Environment"
- DOI: https://doi.org/10.3390/app11104350
- Autores: Huriviades Calderón-Gómez; Luis Mendoza-Pittí; Miguel Vargas-Lombardo; José M. Gómez-Pulido; Diego Rodríguez-Puyol; Gloria Sención; María-Luz Polo-Luque
- Acceso del artículo: Open Access (OA)
- Fecha de publicación: 11 de mayo de 2021
- Idioma: Inglés

2.3.5 Índices de calidad del artículo

- Revista: Applied Sciences-Basel
- ISSN: 2076-3417
- Editor: MDPI
- Volumen y número: 11-10
- Índice de impacto y ranking JCR (2021): 2.838
    - Engineering, multidisciplinary – Q2
- Índice de impacto y ranking SJR (2021): 0.507
    - Computer Science Applications – Q2
    - Engineering (miscellaneous) – Q2

2.3.6 Artículo publicado

applied sciences

MDPI

Article

Evaluating Service-Oriented and Microservice Architecture Patterns to Deploy eHealth Applications in Cloud Computing Environment

Huriviades Calderón-Gómez [1,2], Luis Mendoza-Pittí [1,2], Miguel Vargas-Lombardo [2,*], José Manuel Gómez-Pulido [1,3], Diego Rodríguez-Puyol [3,4,5], Gloria Sención [6] and María-Luz Polo-Luque [3,7]

[1] Department of Computer Science, University of Alcalá, 28805 Alcalá de Henares, Spain; huriviades.calderon@edu.uah.es (H.C.-G.); l.mendoza@edu.uah.es (L.M.-P.); jose.gomez@uah.es (J.M.G.-P.)
[2] e-Health and Supercomputing Research Group (GISES), Technological University of Panama, 0819-07289 Panama City, Panama
[3] Department of Medicine and Medical Specialties, Ramón y Cajal Institute for Health Research (IRYCIS), 28034 Madrid, Spain; diego.rodriguez@uah.es (D.R.-P.); mariluz.polo@uah.es (M.-L.P.-L.)
[4] Foundation for Biomedical Research, Hospital Universitario Príncipe de Asturias, 28805 Alcalá de Henares, Spain
[5] Department of Medicine and Medical Specialties, University of Alcalá, 28805 Alcalá de Henares, Spain
[6] School of Medicine Autonomous, University of Santo Domingo, Santo Domingo 10105, Dominican Republic; dragloriasencion@gmail.com
[7] Department of Nursing and Physiotherapy, University of Alcalá, 28801 Alcalá de Henares, Spain
* Correspondence: miguel.vargas@utp.ac.pa

check for updates

Citation: Calderón-Gómez, H.; Mendoza-Pittí, L.; Vargas-Lombardo, M.; Gómez-Pulido, J.M.; Rodríguez-Puyol, D.; Sención, G.; Polo-Luque, M.-L. Evaluating Service-Oriented and Microservice Architecture Patterns to Deploy eHealth Applications in Cloud Computing Environment. *Appl. Sci.* **2021**, *11*, 4350. https://doi.org/10.3390/app11104350

Academic Editor: Alessandro Di Nuovo

Received: 13 April 2021
Accepted: 6 May 2021
Published: 11 May 2021

Abstract: This article proposes a new framework for a Cloud-based eHealth platform concept focused on Cloud computing environments, since current and emerging approaches using digital clinical history increasingly demonstrate their potential in maintaining the quality of the benefits in medical care services, especially in computer-assisted clinical diagnosis within the field of infectious diseases and due to the worsening of chronic pathologies. Our objective is to evaluate and contrast the performance of the architectural patterns most commonly used for developing eHealth applications (i.e., service-oriented architecture (SOA) and microservices architecture (MSA)), using as reference the quantitative values obtained from the various performance tests and their ability to adapt to the required software attribute (i.e., versatile high-performance). Therefore, it was necessary to modify our platform to fit two architectural variants. As a follow-up to this activity, corresponding tests were performed that showed that the MSA variant functions better in terms of performance and response time compared to the SOA variant; however, it consumed significantly more bandwidth than SOA, and scalability in SOA is generally not possible or requires significant effort to be achieved. We conclude that the implementation of SOA and MSA depends on the nature and needs of organizations (e.g., performance or interoperability).

Keywords: cloud computing; eHealth; elderly people; infectious diseases; MSA; SOA; telemonitoring; versatile high-performance

1. Introduction

Recent developments in the field of Cloud computing have stimulated the need to create new software solutions for the monitoring, management and optimization of an organization's services, in which developers build an application and are not concerned about the underlying infrastructure. For this reason, several authors have proposed different software solutions, such as infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS) and function as a service (FaaS), focused on the intensive processing of data from different sources (e.g., Internet of Things (IoT) devices, data repositories or other sources) that will be consumed by the services to offer users different business-specific functionalities [1–5].

Within this context, the eHealth field is an area that is becoming increasingly important in public health policies because health care delivery systems (e.g., public health management or electronic health records (EHR)) are constantly evolving towards more patient-centered customized services and digital transformation (e.g., eGovernment) [6,7], which, based on data collected by the eHealth systems in conjunction with the recommender systems, can accelerate and drive decision making [8].

Considering the above, we have identified a growing concern regarding the increase in the aging population in our society. In Europe, the expected growth of the population over 65 will be 74% by 2060 [9–11], while in North America, it will be 20% by 2050 [12–14]. This trend places a serious burden on the public health system, since this group of people have high age-related comorbidity (e.g., degeneration in physiology and the abilities of memory, vision and decision), consuming extensive resources of both primary and specialized care [9,15], and consequently, it is more difficult for the elderly to adapt within society.

Telemonitoring systems are very popular today, since these systems are primarily based on the remote analysis of biometric data or daily activities, stemming from either a low availability of medical centers or mobility complications of people. These systems require the use of the Internet of Healthcare Things (IoHT) such as body sensors, bed sensors and smart watches for the collection and monitoring of people through their data, which allows the detection of anomalies in the vital signs of the elderly, and thus alerting medical personnel to these problems [16–19]. In addition, telemonitoring is an effective low-cost means for periodic health exams or preventive treatments in nursing homes [13,20].

Based on the ideas presented and the use of telemonitoring, we have focused on the design, development and implementation of a platform aimed at detection and clinical diagnosis assisted by a recommender system based on artificial intelligence (AI) algorithms within the field of infectious diseases for the elderly population living in nursing homes. It should be noted that this article is part of an extensive study in the field of infectious diseases focused on the three target groups of acute respiratory infections, urinary tract infections and skin and soft tissue infections. However, the article presented here is based in part on several previous studies by the partners of the Design and implementation of a low-cost intelligent system for the prediagnosis and telecare of infectious diseases in elderly people (SPIDEP) [21–25].

Another priority task in the concept of a platform targeted for Cloud computing environments is to define the architectural pattern to be implemented in the platform (e.g., service-oriented architectures (SOA) or microservices architecture (MSA)), which should be based on the needs (e.g., agility, maintainability or high-performance) and the nature of the organization (e.g., eHealth, transport, energy or supply chain management) [7,26–29].

For these reasons, this article describes the work done to evaluate and contrast which of the architectural patterns (SOA or MSA) is more consistent with the attributes of software focused on versatile high-performance within the eHealth context [21], while considering the strengths and weaknesses of each architecture based on quality of service (QoS) and its corresponding metrics. Therefore, it has been necessary to modify the SPIDEP platform into two variants following the guidelines of each architectural pattern, which allows us to analyze which of the variants optimally meet the proposed specifications.

However, the contributions of this article are: (i) to provide an overview of the differences between SOA and MSA; (ii) to identify the implications of designing, implementing and deploying SOA and MSA oriented platforms; (iii) to measure the performance of SOA and MSA variants, taking as reference the quantitative values obtained from experiments, and to contrast these obtained results with respect to existing research results. Nevertheless, the SOA and MSA architectural patterns can be considered complementary allies for an interenterprise architecture that confers a suite of different services rather than being competitors; therefore, it is expected that this future architecture will allow the integration and interconnection of different eHealth applications along with microservices adapted to machine learning for various medical scenarios (e.g., a patient monitoring system for hemodialysis or early forecasting of COVID-19).

This article is organized as follows: Section 2 presents a brief description of the related works, including the motivation for the SPIDEP project and its importance in the field of eHealth. Section 3 details the case study applied in SPIDEP, which is a project aimed at telemonitoring in nursing homes. Section 4 describes in detail the architectural models applied in the designs of the SPIDEP platforms and the technologies implemented. Section 5 shows the results of the spike testing based on the three categories of response time, efficient use of infrastructure and network consumption. The benefits and disadvantages of the results are evaluated in Section 6. Finally, Section 7 provides the conclusions of this study, analyzes its potential and suggests future research activities.

2. Background and Related Work

Currently, there is a growing interest in the use of archetypes for the development of various eHealth applications to represent the structure of clinical information and its specifications within a platform [30–32]; examples include traditional information systems, clinical decision support systems, platforms oriented to the HL7-FHIR protocol or telemonitoring systems, whose purpose is to significantly improve the accuracy of medical diagnosis by establishing expert systems-based prediction approaches or other means of artificial intelligence.

This interest is why we have performed a brief review of the various existing projects that show different analyses, applications and research conducted in this field; however, classifying these projects according to the architectural pattern implemented (i.e., SOA or MSA) was needed since it is necessary to analyze how SOA and MSA influence the development, integration and deployments of eHealth applications and how these patterns adopt principles or practices to address the software requirements.

2.1. Service-Oriented Architecture (SOA)

According to the norms and standards of service-oriented quality [33,34], SOA was one of the most used software architectural approaches in the early 2000s; it was used for the design and development of applications as services. This architecture aims to provide uniformity in the design, implementation and invoking of services required to meet the desired needs of the application [35]; that is, any process, subprocess or logic of an organization can be encapsulated in services [36].

As can be inferred, SOA is a business-focused information technology (IT) architecture that is supported by the integration of services. Consequently, all the services designed in SOA must comply with the following three fundamental software attributes [37,38]: First, each developed component must have the ability to be reused within the application (reusability). Second, the components must be able to communicate with each other or with other external applications (interoperability). Third, the application must allow the ability to adapt to future needs or objectives of the organization (extensibility).

As a follow-up to this activity, several research projects that propose the combined use of this architectural pattern with several software attributes will be outlined below, focused on various eHealth scenarios.

For example, Gavrilov et al. [38] propose a novel conceptual model of EHR focused on the aspect of cross-border interoperability [39], whose objective is to meet the needs of interconnectivity between the different eHealth systems of Macedonia, under the specifications of the European Patients Smart Open Services project [40,41]. Thus, the authors chose to combine the SOA architectural pattern with the use of the data warehouse (DW) and the following three technologies oriented to an extract, transform and load (ETL) process: the interface description (using Web services description language (WSDL) documents), the communication protocol simple object access protocol (SOAP) (using extensible markup language (XML)) and universal description, discovery and integration (UDDI) repositories (allowing users to find services) [38]. They have shown that the proposed model achieves a successful data exchange between the various platforms tested. Additionally, this model allows the classification and indexing of the data, which allows SOA to be able to communicate with different applications other than the predefined DW because each healthcare system is designed, implemented and validated by the HL7 fast healthcare interoperability resources (FHIR) protocol.

Likewise, Amin et al. [39] propose building a framework for the exchange of information among the eHealth systems of Indonesia, whose purpose is to integrate and synchronize data from different (heterogeneous) platforms through web interoperability. This framework was designed under the concept of service-oriented analysis and design (SOAD) [42,43], which is composed of the three main phases of the conceptual view (CV), which illustrates to those involved all the activities that encompass the workflow of the organization; the logical view (LV), which encapsulates all the logic identified by the CV in various software services; and the physical view (PV), which implements the different web layers, including the presentation, application service, domain model and data access layers. The three previous phases focus on a logic based on SOA, whose objective is to process all user information and exchange data among platforms using the RESTful protocol [44]. They validated their framework by successfully relating eHealth systems and integrated entities and the exchange of data and information in the form of an interoperability matrix (IM), so that organizations can use this matrix as a point of reference for adaptation.

Hameedet et al. [45] present a framework oriented to the real-time monitoring of the vital signs of patients (e.g., electrocardiogram, body temperature, pulse rate and oxygen in blood); their framework consists of the combined use of the SOA architecture, rich Internet application and the IoT (Arduino and eHealth sensor shield) [46]. This combined use is necessary because the framework requires the ability to communicate between services and their sensors and to make associations between different medical administrations (e.g., clinics or medical centers); therefore, the implementation of the REST protocol was required to comply with this aspect [47]. They have shown that the proposed framework increases the efficiency of data collection because each service is designed, implemented and validated in an automated manner for the monitoring of a large number of patients.

Silva et al. [48] develop a pilot system called the mHealth application, based on existing practices of IoT and the SOA architecture, focused on telemonitoring of patients. The main objective of this system is to optimize the follow-up services and treatment of arrhythmias or other heart conditions of the patients remotely, which provides coverage to rural or difficult to access areas. This system consists of the four main modules of the monitoring process and patient electronics (main component of the application), body sensor (cardiac arrhythmia detection), caching mechanisms (temporarily stores data in case of network unavailability) and an alert messaging system (sends a warning to health care professionals if an anomaly is detected). They validated their system using a performance assessment and quality of experience (QoE); their results demonstrated a significant improvement in medical care and assistance to patients. In addition, an unexpected result was that the management of medications was more accurate, efficient and less costly for the medical institution.

2.2. Microservice Architecture (MSA)

Currently, there is a growing interest in MSA in both the industrial and research sectors due to its various advantages, including flexibility, agility, infrastructure automation and loose coupling [49–51]. MSA is considered a more refined and simplified version of SOA [33,52], since this architecture responds to the needs of scalability of applications by segmenting the logic of an organization into a series of separate services that are executed as independent processes; that is, it is not necessary to use the same languages, databases or development platforms [21,53,54]. In addition, MSA inherits from SOA the concept of interoperability through the implementation of lightweight mechanisms such as HTTP-based API and others [39,50,55].

Many organizations have recognized the benefits in the migration of their legacy software to microservices-based solutions to enable their applications to evolve, rather than acquiring new third-party software [52,54]. However, performing this process (known as granularity) requires using a pathway or model that can successfully migrate the application; consequently, this factor depends on the use of design patterns (e.g., decompose by business, by verb or use case, by subdomain or by nouns or resources) and how the performance stability of the services is affected [50,51,53,56].

A description of the most significant studies of the new MSA solutions within the eHealth domain is necessary.

Carranza-García et al. [57] present two technological solutions based on MSA (i.e., cognitive training "VIRTRAEL" and frailty prevention "PREFRA"). These solutions aim to integrate the different age-related aspects from the cognitive, physical and social levels; therefore, when addressing these factors, more complete and reliable assessments of the elderly are obtained. However, to achieve this objective, the systems must have the four essential software attributes of reusability, extensibility, interoperability and composition, since these are required for the design, development and implementation of more complex functionalities that make use of different recent technologies (e.g., web or mobile). They have shown that the proposed technologies are a viable solution aimed at meeting the required quality attributes.

García-Moreno et al. [58] develop an MSA-based system that collects sensory data of the elderly in their daily activities (e.g., toilet hygiene or functional mobility); these data include not only a physical dimension but also cognitive and social dimensions. This system is supported by using IoT (i.e., wearable devices), which are used by microservices focused on the collection, analysis and interpretation of data. In addition, this system is supported by several machine learning algorithms (e.g., kNN, RF or NB) to build more accurate models for detecting anomalies in people [59].

Jarwar et al. [17] provide a model focused on semantic interoperable data-driven microservices. This model is responsible for capturing, representing and analyzing various types of data related to depressive disorders (DD). These data are vital to providing the model with a way to monitor the symptoms of DD through the use of indicators (e.g., user facial expressions, voice tone, user emotions from wearable sensors, social network services posts (SNS) and tweets). Generally, microservices do not make intelligent decisions based on data [60]; therefore, it is necessary to implement the following two dedicated servers in the model for this scenario: data mining/machine learning (extracts features and monitors the symptoms from the sensor data and SNS) and web of objects server (applies the semantic web technologies and inference of the user situation to provide the services to the users).

Da Silva et al. [30] propose a tool called Microservice4EHR, which is based on the OpenEHR standard used for the computational representation of EHR (archetype) [61] and the conceptualization of a microservices-oriented software architecture. This tool consists of the five functionality phases of a tool access key, the host server address of the MSA component, a graphical interface with health data based on archetypes, a graphical interface filename and the input/output data (in JSON format), which is processed and presented in a web application [30]. Moreover, all these approaches have the purpose of building reusable components that can be used as building blocks in health applications (e.g., a blood donation center in northeastern Brazil). Additionally, this attribute improves the capacity of the maintainability and interoperability of the applications between the information systems.

2.3. SOA vs. MSA

Considering the above, the previously cited studies reveal different eHealth applications, based on SOA or MSA, that focus on one or several of the software attributes of interoperability, maintainability, reusability and scalability.

Both architectures have their strengths and weaknesses, according to the needs of the software and implementation [62]. SOA focuses mainly on the sharing of services in relation to the processes or capabilities of an organization [50,63]. In this sense, SOA can refer to a broad accumulation of knowledge over the last two decades [62]. Regardless, this approach presents many issues when correcting errors or adding new functionalities, since the design of the application is very complex and time consuming. In addition, scalability in SOA is generally not possible or requires significant effort to be achieved. Consequently, organizations are beginning to migrate to other architectural solutions such as MSA [64,65].

MSA allows efficient development, implementation and maintenance of services compared to SOA. In fact, since the services are autonomous, they can be developed, maintained and tested independently, which facilitates their automation and continuous deployment (e.g., DevOps practices) [21,50,66]; however, such practices entail complexities inherent to a distributed computing environment in relation to the isolation, availability, security and transaction of services [33,50], which in turn causes a fundamental problem known as granularity (i.e., if a microservice needs to decompose or merge) [21,51].

Based on these considerations, the implementation of SOA or MSA depends on the nature of the target system to be developed. Thus, for the development of our eHealth application, it is of utmost importance to contrast the performance between SOA and MSA based on the concept of versatile high-performance [21], since it is necessary to meet the needs of the environment of health organizations, focused on horizontal scalability. Quicker and more accurate real-time responses to demand peaks are thus enabled [22]. Additionally, it is necessary to allow the hierarchy and consolidation of a patient's clinical information, including what is necessary not only for generating standard reports but also for allowing the integration of recommender systems, to support decision making with consistent and truthful parameterized indicators of the data collected [21].

3. A Case Study from eHealth

The work presented here is based on a previous study performed by the SPIDEP project part of the Joint Call ERANET LAC 2015—FP7 [67], whose objective was to build an intelligent system based on information and communication technologies (ICT) to support the early diagnosis of infectious respiratory and urinary diseases in the elderly [21,24]. For this, an international consortium was formed between the clinical research teams and the Infectious Diseases Unit of the Hospital Universitario Principe de Asturias (Alcalá de Henares, Spain), the Chronic Diseases and Cancer Area of the Ramón y Cajal Institute for Health Research (Madrid, Spain) and the School of Medicine Autonomous University of Santo Domingo (Dominican Republic) and the research teams of the Computer Sciences group of the University of Alcalá (Universidad de Alcalá) and the Technological University of Panama (Universidad Tecnológica de Panamá).

This study incorporates patients living in the Francisco de Vitoria and Cardenal Cisneros nursing home in Spain and residents of the Carls George and Nuestra Señora del Carmen Institution for the Care of the Elderly in the Dominican Republic. The main purpose of the study is to provide health services and care to people at risk or who have difficulties in accessing health services through the use of new technologies to achieve greater efficiency in the organization and care of special population groups, with the added value of cost savings [21].

Actually, health monitoring has become one of the biggest areas in the technology industry, developing many smart devices like watches or another more specific body sensors. This has allowed researchers to contribute to the development of eHealth platforms focused on regularly monitoring the vital signs of at-risk patients, with the aim of carrying out treatments or preventive interventions to minimize health problems and reduce emergency care [10,16–18,46,48,58].

4. Model and Design of SPIDEP Platforms

To perform this study, a merging of the computational paradigms of the Edge and the Cloud was used [68,69], based on a hybrid Cloud architecture to support medical telemonitoring applications [21] under the two architectural patterns (i.e., SOA and MSA).

In this sense, we focus on the evaluation of the design, development and implementation of the services created for our use case (SPIDEP) and how they affect performance related to the response time to queries, the efficient use of infrastructure and network consumption.

4.1. Analysis of Software Attributes

This preliminary step is fundamental for the design and development of applications, since it is necessary to define an adequate set of software attributes according to the needs identified [7].

In response to these considerations, it is of utmost importance to mention our needs; they are as follows: (i) neutral technology—those responsible for the development, implementation and monitoring of the assigned services may decide which is the best technology to use according to their objectives; (ii) automation and continuous deployment—each component should be autonomous, independent and focused on a specific task according to the established decomposition (e.g., by verb or use case). In addition, it must have the individual ability to replicate to balance the load, as necessary; (iii) scalable to the demand—the services require a horizontal scalability that allows a faster and more precise reaction to the demand peaks; (iv) interoperable—each service must run a lightweight communication mechanism (e.g., RESTful) to communicate between the different applications or services, either their own or those of third parties. In addition, this mechanism should not have access to visual elements, i.e., it must be separate from the front-end; and (v) high performance—the application must support a large volume of queries of the REST methods (e.g., GET, POST, PATCH and DELETE) from the integration, transfer and storage of the biometric sensors from the different nursing homes (Edge) to the various services offered by the SPIDEP hybrid (Cloud).

As explained in Section 2.3 (SOA vs. MSA), it was decided to adopt the concept of versatile high-performance, since it allows us to meet the needs of the project [21].

4.2. Variants of Software Architectures

Considering the above, it was necessary to build two variants of SPIDEP, according to the specific attributes of each architectural pattern; however, SOA and MSA have different approaches in terms of how their services are designed, implemented and deployed [33,70]. In addition, it must be remembered that MSA remains an architecture in training, while SOA has been studied for more than a decade [35,62].

Therefore, it has been decided to establish two branches of the SPIDEP application since it allows us to contrast and evaluate which of the variants fully meets our criteria [71–73].

4.2.1. SPIDEP SOA Variant: Platform Design

The first proposed platform (SPIDEP-SOA) is intended to manage the interactions between users (consumers in the upper layer) and health services (software functionalities in the lower layer), according to the changing needs of the platform (technology, demand and security), as shown in Figure 1.

Next, we will briefly outline our first variant of the platform, which consists of six layers focused on the management of services under the computational paradigms (i.e., Edge and Cloud) [23], as follows:

- Consumers: This layer allows interaction and exchange between the health professionals of the different homes (nurses and assistants), the hospitals in charge of each region (general physicians) and the eHealth applications (SPIDEP or others).
- Edge layer: Establishes HTTP connections with the APP for the asynchronous sending of medical data in the Cloud (i.e., the Cloud layer) [33], whose data come from the different users and their sets of biometric sensors (i.e., blood pressure, pulse rate, body temperature, oxygen saturation and electrodermal activity) [23]. It should be noted that this communication is encrypted by the security layer.
- Cloud layer: This layer performs heavyweight computations; that is, it helps consumers discover, route and deploy health services and infrastructure. In addition, this

layer consists of the three sections of infrastructures and resource, support functions and control system. Additionally, different algorithms are used to achieve various objectives (e.g., error tolerance and load balancing) [74].

- Service layer: Contains services for the end users like medical personnel and nursing home administrative staff with their respective profiles.
- Security layer: Ensures access control and consumer authentication through established security policies, whose objective is to determine the user privileges for certain resources and/or specified levels of services.
- Management system: This layer mainly controls the flow of messages between the different layers. In addition, it is responsible for executing the real-time adaptation of the services (versatility) according to demand, e.g., automatically adding a new instance, monitoring the status of the different components that integrate the platform or other monitoring tasks.

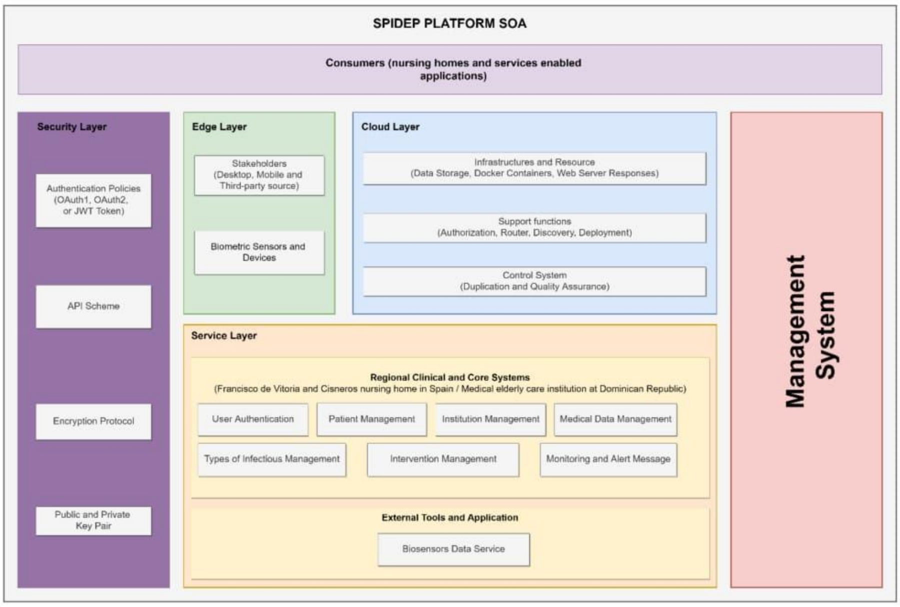

Figure 1. General scheme of the services applied in the SPIDEP platform under the SOA variant (SPIDEP-SOA).

It should be noted that this variant of SPIDEP was a refactoring of the legacy application [23] whose purpose was the modernization of existing services oriented to our current software attribute (i.e., versatile high-performance) [64]; however, the strengths and weaknesses of SOA must be considered.

Considering the above, the strengths of SOA are the following:

- Provides a higher level of compression related to the abstraction of the interfaces than MSA (decoupling) [62], since SOA traditionally has the integration of an enterprise service bus (ESB) [75], which facilitates the interaction between the services of the platform.
- Offers extensive knowledge that has been established over more than a decade, which has proven to be effective and reusable in building platforms. It is thus easier for developers to ensure the quality of service required by the organization [76].

- Suitable for large and heterogeneous systems consisting of many applications, non-independent services and shared components [33,77]. It provides a roadmap for the adoption of its principles, which allows developing or transforming the capabilities of an organization's software system into reusable services for greater flexibility and agility [78].

However, when applying SOA, the following weaknesses or difficulties must be considered:

- Message exchange is traditionally synchronous (wait-to-connect); i.e., it depends on the state of the ESB [33]. However, when applying a design oriented to service implementation patterns, the support of asynchronous messages is integrated, increasing complexity and maintainability and reducing flexibility [33,76,79,80].
- When developing platforms based on SOA, it is complex to add or modify functionalities to what has already been established; redesigning this type of platform consumes significant time and resources [64,81].

4.2.2. SPIDEP SOA Variant: Implementation of the Platform

For the implementation of the SPIDEP-SOA platform, we opted to modernize the first version developed (Beta "traditional implementation of seven SOA services, without API support") [23] to the release candidate (RC) ("Modernization of services and current technologies (language, database, libraries and integration of an API)"), whose purpose is to provide the necessary attributes for the implementation of versatile high-performance applications and contrast it with the MSA variant. Therefore, we proceeded to the refactoring and restructuring of the existing components along with their data [53,82], as shown in Table 1 for each version.

Table 1. General structure in the SPIDEP-SOA platform.

Description		Version	
Category	**Subcategory**	**Beta**	**RC**
System architecture	Architecture styles	SOA	SOA-based on Quality Requirements
	SOA patterns	Enterprise Service Bus (ESB)	Service Implementation Patterns
	Communication protocol	HTTP	HTTP
	HTTP methods	Disable	GET, POST, PATCH & DELETE
	Messaging type	Synchronous	Synchronous/Asynchronous
Used technologies	Programming language	PHP 5.6	PHP 7.4
	Framework	Laravel 5.1 LTS	Laravel 6 LTS
	Core libraries	acoustep/entrust-gui; zizaco/entrust; Laravel Passport; mockery/mockery; phpunit/phpunit; tinker; composer, and others	Guzzle; fideloper/proxy; kylekatarnls/laravel-carbon-2; zizaco/entrust; Queues; Laravel Sanctum; mockery/mockery; phpunit/phpunit; tinker; composer, Custom libraries, and others
	UI	Bootstrap 3	Bootstrap 4
	Web server	Apache	Apache as an endpoint and NGINX as a load balancer for the implementation
	DBMS	MariaDB 10.1	Postgresql 10
	Data access methods	Stored procedures	Stored procedures
	Object–relational mapping	Eloquent	Eloquent
	API Scheme	Disable	RESTful (JSON Request/Response HTTP)
	API Gateway	Disable	Central Entry Point
	Deployment	Bare Metal	Custom Docker Container
	OS	Ubuntu 16 LTS	Ubuntu 18 LTS
	Authentication Scheme	HTTP Basic Authentication	Custom Authentication (OAuth1, OAuth2 or JWT Web Token)
	Encryption Protocol	OpenSSL (AES-256 and AES-128)	OpenSSL (AES-256 and AES-128) and public/private rsa key pair

4.2.3. SPIDEP SOA Variant: Deployment of the Platform

When applying an SOA design pattern (compound, service implementation, service composition, inventory and others) [80], a positive impact is obtained on certain quality requirements (i.e., performance or interoperability), which negatively affects other quality requirements (i.e., redundancy or security). Therefore, it is important to select design patterns according to the software attributes and quality requirements [76].

Considering the above, service implementation patterns were chosen for the development of the SPIDEP-SOA platform, since they allow creating versatile and scalable services that run in the Cloud [83,84]; however, an automated workflow must be defined for the integration and continuous deployment of services [21,80].

Within this concept, we have adjusted our framework for this variant (mainly in the first three sections) [21], because SOA is geared to a modular design to reduce dependencies between platforms [37,85], i.e., that the services collaborate and share their resources (loose coupling), unlike MSA where the services are decoupled and isolated [33].

In addition to the situation, a general overview of the SOA-based proposal follows [21]:

- Teamwork: Responsible for the coordination and monitoring of all platform services (business-centric IT) [86].
- Services: Unlike our MSA framework, each team does not have the autonomy to decide which is the best technology for the development of services. These teams should adhere to the Teamwork decisions that follow the business functionalities. For example, all SPIDEP-SOA services were developed under the Laravel PHP framework and a single DBMS, PostgreSQL 10 (the core systems, except for the mobile app, were developed in Java); however, all services are supported by an asynchronous HTTPS server that is based on transport layer security (TLS) for client authentication and can replicate instances based on demand, unlike the traditional SOA approach.
- Code repository: A Git repository is used to maintain version control of the code from the development branch of the services that make up the platform [87,88].
- Software analysis and testing: The source code and the environment are reviewed to ensure correct functionality and that they meet the desired standard; then, unit tests of the services are performed (e.g., PHPUnit).
- Docker containers (test environment (TE)): To ensure the stability and scalability of our platform, we opted for the customization of a test container (Ubuntu Server 18.04 LTS, 2 Core, 4 GB RAM and 80 GB SSD); however, these containers must pass preliminary test criteria defined by Teamwork before being published in the Docker Hub in a private repository.
- Quality assurance (QA): Functional tests, interoperability tests, and load and performance tests are performed to ensure the quality of services [21,34].
- Docker containers (production environment (PE)): If the integration of all services in the platform is successful, the stable version of the container is deployed to Docker instances (under the same TE attributes); otherwise, the code should be debugged and the QA tests rerun.
- Performed postproduction: Additional tests (e.g., long-term spike testing) are run to ensure that the services work properly in all instances [21,53,89].
- Managed Kubernetes: Kubernetes is used for automating management of computerized services, since it reduces the manual and repetitive processes involved in container management [90]. In addition, Kubernetes scales according to demand, i.e., it increases the use of resources in high demand, and if demand decreases, idle resources are reused [91]. Unlike the traditional SOA approach (which does not have scalability support), it was necessary to make several adjustments to the RC version of the platform.

4.2.4. SPIDEP MSA Variant: Platform Design

The second proposed platform (SPIDEP-MSA) is geared to the use of microservices in the expected medical scenarios of infectious diseases (like its other variant) [21,22,24], as shown in Figure 2.

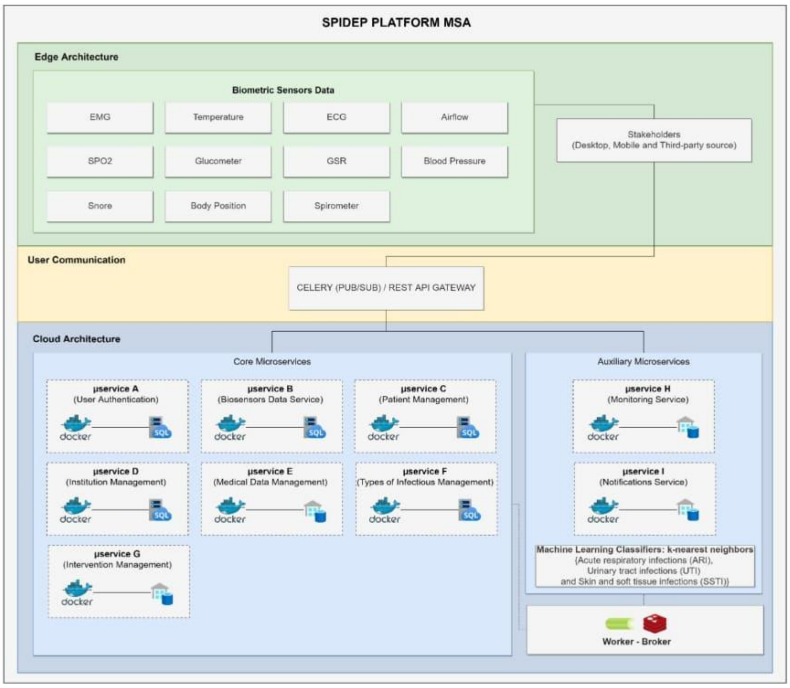

Figure 2. General scheme of the services applied in the SPIDEP platform under the MSA variant (SPIDEP-MSA).

Both platforms aim to support the interpretation of changes in vital signs of elderly people living in residential facilities, thus alerting the medical team in advance of possible infections; however, the MSA variant has a recommender system to improve decision support in the prediagnosis of infectious diseases [25,92]. Following, we will briefly explain the flow of the SPIDEP platform in relation to microservices [23]:

- Edge architecture: Manages the collection, preprocessing and sending of data from the different biometric sensors (e.g., ECG, blood pressure, SPO2 or others), whose data are backed up locally. If necessary, temporary actions can be performed (e.g., delete, filter or update) before the data are sent to the Cloud architecture, which enables analysis with local data to streamline decision making in case of emergency [16,93]. It should be noted that the connections with each APP are made asynchronously and independently for each stakeholder.
- User communication: Provides the necessary mechanisms for correct communication between the different layers and sublayers (e.g., data, validation of credentials or other). This communication is achieved through the representational state transfer (REST) protocol and its methods (i.e., GET, POST, PATCH and DELETE); however, this protocol requires the use of an API gateway. Therefore, taking advantage of some of the strengths of MSA (e.g., decoupling and isolation of microservices) [30], the

Backends for Frontends design was implemented in the API gateway [56,89]. Requests are handled by each stakeholder (i.e., desktop, mobile and third-party source) and not traditionally (i.e., single entry point); consequently, additional security mechanisms must be implemented for their validation [7,94,95].

- Cloud architecture: Provides services for end users (e.g., core or auxiliary microservices) according to their needs and the levels of user roles and permissions [21]. In addition, it manages the stored data of the different DBMS (database per service) [54,96] to provide benefits to target users (e.g., patients, medical personnel or health authorities) [16].

Similarly, we must remember the strengths and weaknesses of MSA. The strengths of microservices are the following:

- Each microservice is incapsulated; therefore, it has more flexibility to use new frameworks, libraries, data sources and other resources [64].
- It allows horizontal scaling of the services instead of vertical scaling as in the case of a traditional or monolithic SOA, which facilitates the ability to use more computing resources (e.g., CPU, GPU and RAM) that will act as a single unit; however, it can be distributed through multiple virtual networks [50,55].
- Each microservice is autonomous, has independence in the execution of its tasks, can be hosted in a specific server on the Internet, is available on the Internet (i.e., any software can interact with it) and does not share the same resources [30,97,98].
- The fault isolation can be obtained without interrupting the normal functioning of the whole system [16,52].
- However, when applying MSA, the following weaknesses or difficulties must also be considered:
- MSA provides many different advantages, but it also increases a solution's complexity. To address increased complexity, an organization should prepare the specific infrastructure for microservices (i.e., obtain a clear picture of the data structure) [64,99].
- If the level of granularity is determined before knowing the business process or the useful life of the platform, it can lead to problems in the reasoning about the services (e.g., if a microservice should be decomposed or merged with another) [33,51].

4.2.5. SPIDEP MSA Variant: Implementation of the Platform

Similarly, for this variant (SPIDEP-MSA), the respective modernization of the latest built version was performed (β v2 "implementation of the nine microservices, with hybrid instances in MariaDB Galera Cluster and NoSQL, and two EndPoint (Mobile/PC)" [21] to the RC ("Modernization and restructuring of microservices and current technologies (language, database, libraries and integration of a third EndPoint)").

The following is a general description of the technologies implemented in each version, as shown in Table 2.

Table 2. General structure in the SPIDEP-MSA platform.

Description		Version	
Category	**Subcategory**	**Beta v2**	**RC**
System architecture	Architecture styles	MSA	MSA
	MSA patterns	Service instance per-container; Database per service & API Gateway	Service instance per-container; Database per service & API Gateway-
	Communication protocol	HTTP	HTTP
	HTTP methods	GET, POST, PATCH & DELETE	GET, POST, PATCH & DELETE
	Messaging type	Synchronous/Asynchronous	Synchronous/Asynchronous

Table 2. *Cont.*

Description		Version	
Category	**Subcategory**	**Beta v2**	**RC**
Used technologies	Programming language	PHP 7.2 (μs-A) & Python 3.6 (μs-B~I)	Python 3.7 (μs-A~G) & Python 3.8 (μs-H~I)
	Framework	Laravel 5.4 LTS (μs-A) & Django 2.1 (μs-B~I)	Django 2.2 LTS (μs-A~G) & Flask 1.1 (μs-H~I)
	Core libraries	PHP/Laravel dependencies (Guzzle;Laravel Sanctum and others), Python/Django dependencies (setup-tools;requests;psycopg2; requests_futures;lxml;six;django-cas-server;django-cas-ng;djangorestframework; celery, TensorFlow and others) & Custom libraries	Python/Django~Flask dependencies (setuptools; requests; psycopg2; requests_futures; lxml;six;werkzeug django-cas-server;django-cas-ng; python-cas; djangorestframework; Flask-RESTful; celery, TensorFlow and others) & Custom libraries
	UI	Bootstrap 3	Bootstrap 4
	Web server	Apache as an endpoint and NGINX as a load balancer for the implementation	Apache as an endpoint and NGINX as a load balancer for the implementation
	DBMS	MariaDB Galera Cluster (Galera 3), Cassandra & Redis	Postgresql 10, Cassandra & Redis
	Data access methods	Stored procedures & Column Family	Stored procedures & Column Family
	Object–relational mapping	Eloquent, QuerySet & Django-Cassandra-Engine	QuerySet, SQLALchemy & CQLAlchemy
	API Scheme	RESTful (JSON Request/Response HTTP)	RESTful (JSON Request/Response HTTP)
	API Gateway	Backends for Frontends (2 EndPoints)	Backends for Frontends (3 EndPoints)
	Deployment OS	Custom Docker Container Ubuntu 18 LTS	Custom Docker Container Ubuntu 18 LTS
	Authentication Scheme	Custom Authentication [OAuth1, OAuth2 or JWT Web Token] & CAS Protocol 3.0 Specification	Custom Authentication [OAuth1, OAuth2 or JWT Web Token] & CAS Protocol 3.0 Specification
	Encryption Protocol	HTTP Authentication (Tokens and SSL); PBKDF2 with a SHA256 hash; public/private rsa key pair and custom authentication settings	HTTP Authentication (Tokens and SSL); PBKDF2 with a SHA256 hash; public/private rsa key pair and custom authentication settings

4.2.6. SPIDEP MSA Variant: Deployment of the Platform

To meet the software attributes and quality requirements, it was necessary to establish an automated workflow for the integration and continuous deployment of the microservices in SPIDEP-MSA. Therefore, we assigned a cross-team (typically a small group of six to eight people [100,101]) for each case (e.g., decomposed by verb or use case [56]); each team is responsible for the development, debugging and deployment of the assigned microservice. This workflow reduces the time between updates or new system functionality (without affecting end users) and implementation of changes to the production environment (without affecting system performance) [21,102,103].

Considering the above, we implemented the same framework proposed for the MSA variant of SPIDEP (beta v2), whose framework is divided into the following nine phases: (i) cross-team project, (ii) service, (iii) code repository, (iv) software analysis and testing, (v) test environment, (vi) quality assurance, (vii) production environment, (viii) performed postproduction and (ix) managed Kubernetes [21].

Unlike the SPIDEP-SOA variant, this platform requires applying the implementation patterns based on MSA (i.e., service instance per container, database per service and Backends for Frontends), since it is necessary to fully support a horizontal scalability of the resources. However, this entails an increase in the effort and complexity to manage all interactions between microservices, according to the size of the organization to be implemented [60]. Additionally, in this variant, we verify the identity of the user for each message sent and/or received through the signal mechanisms; therefore, a secure communication channel is established for different computer attacks (e.g., man-in-the-middle) during active sessions [104].

5. Validation of Results

5.1. Experiments Settings

To evaluate the performance of the resources of each SPIDEP variant incorporating SOA and MSA (i.e., networks and infrastructure), spike testing was used to obtain quantitative values regarding the behavior of the services and their interaction with end users [53,89,105,106].

Before performing these controlled tests, it was necessary to create two Kubernetes environments (K8s). The first environment, named SPIDEP-SOA-K8s, contains all the services of the SPIDEP SOA variant platform, with the following specifications: a custom instance of an Ubuntu Server 18.04 LTS (2 Core, 4 GB RAM and 80 GB SSD), PostgreSQL cluster (database dedicated server only) and a PODs replica. The second environment, named SPIDEP-MSA-K8s, contains all the services of the SPIDEP MSA variant platform, with the following specifications: each microservice has a custom instance of an Ubuntu Server 18.04 LTS (2 Core, 4 GB RAM and 80 GB SSD), PostgreSQL cluster (database dedicated server only) and no PODs replica. Additionally, each environment uses Apache as an endpoint and Nginx as a load balancer. The various user requests are received initially by Nginx, which sends the request to the corresponding services to address the URI-request [21].

After deployment of the K8 environments, the use of scripts with random variables developed in Apache JMeter (5.2) is required to measure and contrast the performance among the 50 virtual users (simulating the terminals of the biometric sensors) and the SPIDEP platforms; however, to avoid altering the results, it is necessary to use dedicated servers to perform these tests. Therefore, we have selected the BlazeMeter servers (US East [Virginia, Google]) to generate a heavy workload towards the platforms, according to small or medium instances of medical telemonitoring environments and their specifications (approximately 200,000 queries every 20 min) [7,21,53,60,93]; however, these platforms have the capacity of automatic scaling that will determine the distribution of computer resources according to demand (for these tests they were disabled, since we had medium instances), as shown in Figure 3.

It should be noted that these experiments were evaluated and agreed upon by the experts within the project to be implemented in custom instance of an Ubuntu Server 18.04 LTS [21], as they were previously used to perform load and performance tests, all in order to ensure the quality of the services developed (i.e., performance, scalability and availability), and in conjunction with the use of recent technologies such as: (i) Docker, all platform services have been encapsulated as Docker images to enable their simple management, update and deployment processes within a Cloud environment. In addition, the Docker implementation can be based on individual containers or grouped into a combination of elements that form an overall service [28,29]; (ii) Kubernetes, this technology is used for the orchestration of Docker containers, as it allows the initialization and scaling of container-based jobs, the exposure of services, as well as the rescheduling of failed jobs and long running services [90,91].

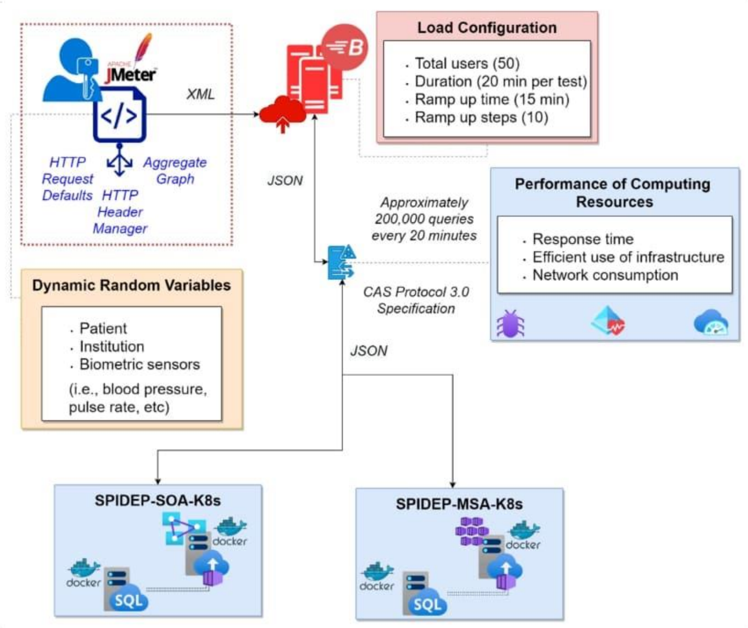

Figure 3. Process Flowchart for the simulation performed on the K8s SPIDEP environments.

However, all this set of technologies allows its adaptation and implementation in a simpler way by developers, which results in consistent, measurable, and replicable results. However, this cloud computing-based framework is intended to be implemented in different Linux distributions (e.g., Fedora, CentOS, Amazon Linux, Debian, and others) or Windows Server (e.g., 2016 or 2019). e.g., 2016 or 2019), since this framework is governed by the segmentation of an organization's logic into a series of separate services that run as independent and isolated processes; that is, it is not necessary to use the same languages, OS, database, or development platforms; however, to replicate these results all the specifications implemented in each variant must be taken into account, for more details please refer to Tables 1 and 2 in Section 4.

Other considerations to take into account would be: (i) type of input and output, for both platforms the RESTful protocol is used to communicate and HTTP methods (GET, POST, DELETE and PATCH). However, the data sent or received are delivered in JSON (Javascript Object Notation) format, since this format is easier to be interpreted by developers or electronic platforms. However, these platforms support other formats like HTML, YAML, XML, XLSX or TXT; (ii) general accessibility, the application was deployed in a cloud environment that can be accessed through a browser (i.e., Google Chrome, Fire-fox, Opera or others); (iii) number of tests performed, four joint tests were created divided into two concepts (Response time and Network consumption, and Efficient use of infrastructure) with an average of 200,000 samples per test for 20 min, grouped in two groups (SPIDEP-SOA-K8s-T1/T2 and SPIDEP-MSA-K8s-T1/T2).

It should be noted that both environments follow the same steps to summon their services. The first step is to authenticate all active sessions using the JWT token that will be verified by the API gateway (all user interface "UI" use is excluded). The second step is to generate approximately 150,000 initial data points in each database, using as seed the existing medical data (8500 records). In this sense, all the data generated is only intended to generate a workload that will drastically increase (ten of ten users every five minutes) to emulate a specific number of concurrent user sessions (50 virtual users); it will not be used to train, validate or test any expert system. The third step is to randomly execute an HTTP command (e.g., GET, POST, PATCH and DELETE) in the specified service. We have chosen the two most demanding services (approximately 75,000 consultations per session) for both platforms (medical data management (service E) and intervention management (service G)). Services H and I are excluded because of their incompatibility in SPIDEP-SOA; they are in the experimental phase of being integrated with SPIDEP-MSA. The fourth step is to monitor and record all the queries made in each test.

5.2. Evaluations and Results

To evaluate these experiments in SPIDEP, a total of 800,000 queries were sent for a period of 80 min (approximately 200,000 queries every 20 min). It should be noted that these values are obtained from the experiments performed between the two variants of SPIDEP, for more details please refer to Table 3. Another factor to consider is that 50 users are simulated requesting a set of random data simultaneously. Now, the response times of the instances are acceptable and are sufficient for this number of users; however, it must be kept in mind that this quality degrades as the number of users increases, if not foreseen with an adequate scalability of the infrastructure vs the demand [21].

The results were collected through a tabular output report generated after each load test, corresponding to its environment, as shown in Table 3. For the tables, the following labels are used: SPIDEP-SOA-K8s-T1 is a K8 environment that hosts all the services of the SPIDEP-SOA platform, this test will run multiple HTTP queries to a single service (service E); SPIDEP-MSA-K8s-T1 is a K8s environment that hosts all the services of the SPIDEP-MSA platform, this test will run multiple HTTP queries to a single service (service E); SPIDEP-SOA-K8s-T2 is a K8 environment that hosts all the services of the SPIDEP-SOA platform, this test will run multiple HTTP queries to two services simultaneously (services E and G); and SPIDEP-MSA-K8s-T2 is a K8 environment that hosts all the services of the SPIDEP-MSA platform, this test will run multiple HTTP queries to two services simultaneously (services E and G).

The report contains several important values divided into four labels reflecting the type of environment executed, including the type of HTTP method executed, the number of samples, average response time, number of requests that are processed (i.e., hits per seconds), 90th percentile, 95th percentile, 99th percentile, number and percentage of failed requests, the average latency time and the data consumption transferred between the user and the service.

Additionally, several stress simulation tests in the infrastructure that host both variants of SPIDEP were performed, whose functions are to detect any problems that may arise in the platform. Reviewing CPU performance, memory, network I/O and connections (e.g., engine health to BlazeMeter) indicates whether the SPIDEP infrastructure itself is capable of supporting the demand-related bottlenecks or errors that appear [21,53], as shown in Figures 4–7, with one figure for each environment.

Table 3. General results of the simultaneous queries made to the RESTful methods of the SPIDEP application, under the SOA and MSA architectural styles, invoking single and dual services (i.e., medical data management (service E) and intervention management (service G).

Labels	HTTP Methods		Samples	Avg. Response Time *	Avg. Hits/s	90% Line *	95% Line *	99% Line *	Error Count and Percentage	Avg. LATENCY *	Avg. Bytes (Kbytes/s)
SPIDEP-SOA-K8s-T1	GET		25,245	395.2	21.04	607	631	711	0 (0%)	395.19	12.25
	POST		25,237	401.26	21.03	615	639	719	0 (0%)	401.25	4.87
	PATCH		25,226	388.84	21.02	603	627	707	0 (0%)	388.83	4.87
	DELETE		25,217	390.3	21.01	603	627	723	0 (0%)	390.29	4.86
SPIDEP-MSA-K8s-T1	GET		27,838	344.38	23.2	543	579	663	0 (0%)	344.34	22.99
	POST		27,830	363.9	23.21	567	599	691	0 (0%)	363.87	9.74
	PATCH		27,813	362.23	23.20	571	603	683	0 (0%)	362.19	9.98
	DELETE		27,805	358.68	23.19	563	595	683	0 (0%)	358.65	9.98
SPIDEP-SOA-K8s-T2	Intervention	GET	50,511	390.89	42.09	603	631	715	0 (0%)	390.88	16.2
	Management	PATCH	50,346	387.13	41.95	599	627	711	0 (0%)	387.12	9.65
	Medical Data	POST	50,487	390.39	42.01	615	639	719	0 (0%)	396.39	9.74
	Management	DELETE	50,947	390.12	42.45	603	627	695	0 (0%)	390.11	9.79
SPIDEP-MSA-K8s-T2	Intervention	GET	102,009	389.72	85.01	599	639	715	0 (0%)	335.65	57.68
	Management	PATCH	108,978	364.84	90.82	579	623	699	0 (0%)	364.84	36.35
	Medical Data	POST	118,534	335.37	98.78	535	571	639	0 (0%)	384.02	42.51
	Management	DELETE	112,221	354.22	93.52	555	595	683	0 (0%)	354.21	36.63

* Time measurements are averages in ms.

Figure 4. Engine Health Report SPIDEP-SOA-K8s-T1. The colored lines represent various parameters as follows: the blue line is the bandwidth traffic; the purple line is the memory load generated by the users; the green line is the CPU load generated by the users; and the red line is the active connections within the test.

Figure 5. Engine Health Report SPIDEP-MSA-K8s-T1. The colored lines represent various parameters as follows: the blue line is the bandwidth traffic; the purple line is the memory load generated by the users; the green line is the CPU load generated by the users; and the red line is the active connections within the test.

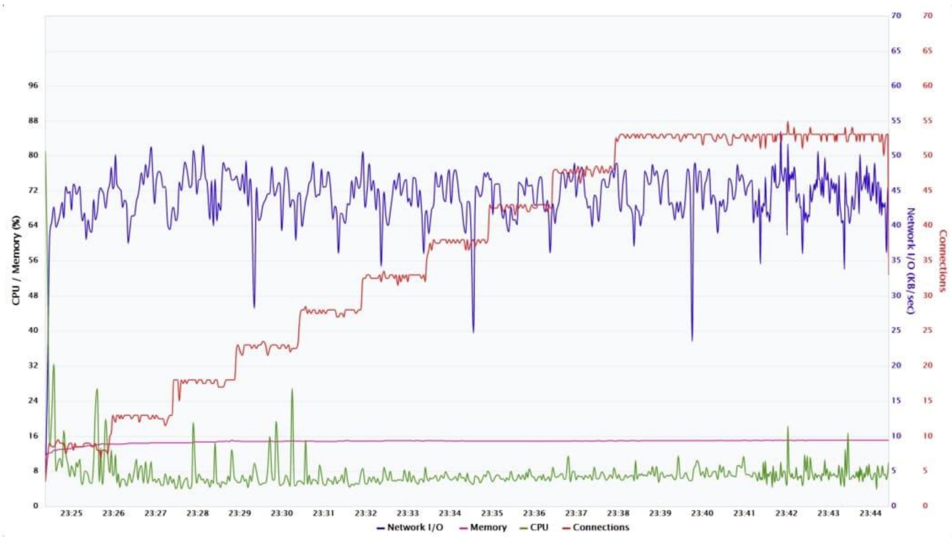

Figure 6. Engine Health Report SPIDEP-SOA-K8s-T2. The colored lines represent various parameters as follows: the blue line is the bandwidth traffic; the purple line is the memory load generated by the users; the green line is the CPU load generated by the users; and the red line is the active connections within the test.

Figure 7. Engine Health Report SPIDEP-MSA-K8s-T2. The colored lines represent various parameters as follows: the blue line is the bandwidth traffic; the purple line is the memory load generated by the users; the green line is the CPU load generated by the users; and the red line is the active connections within the test.

6. Discussion

The purpose of this study is to evaluate and contrast the performance of computing resources (i.e., response time, efficient use of infrastructure and network consumption) used by the different services for each SPIDEP variant, based on the concept of versatile high-performance (acceptance criteria) and the quantitative values obtained from the various tests (performance testing). Another purpose is to offer a software solution focused on the strengths and weaknesses of each variant (SOA and MSA) to provide customized eHealth functionalities and host functionalities based on AI algorithms (e.g., recommender system based on deep learning), according to the storage of adequate data for remote medical care scenarios (telemonitoring) and the demand and the size of the instances of the medical organizations (e.g., clinics, nursing homes or hospitals).

There are several significant findings from this study. First, it is important to consider the acceptance criteria such as the average response time of the queries, the 90th percentile, and their relationship with the percentage of failed requests. Therefore, according to various investigations [107–110], it was established that the acceptable limit (criterion A) for the average response times of the queries and the 90th Percentile is 5000 ms (five seconds) and the acceptable limit (criterion B) for the percentage of failed requests is 5%. Any test with results beyond the established limits is considered to have unacceptable performance [21].

Considering the above, from Table 3 we can establish that the tests with 50 concurrent users are acceptable, since both variants meet the two criteria defined; however, we have an interesting case in the SPIDEP-SOA-K8s-T2 and SPIDEP-MSA-K8s-T2 environments, specifically in the parameters of total number of queries, average response time and network consumption. As shown, in the SPIDEP-SOA-K8s-T2 environment, there is an average of 50,572 queries made (42.14 avg. hits/s), while SPIDEP-MSA-K8s-T2 has an average number of 110,435 queries made (92.02 avg. hits/s). This remarkable difference is due to the following two factors:

- Architectural style: The attributes of MSA focus more on supporting the streamlining and reduction of microservice deliveries in the shortest possible time (all microservice must be lightweight, decoupled, isolated and independent of any programming language, libraries or databases). Conversely, SOA makes use of the ESB or central entry point, which are not considered agile enough [33] because the interactions between the services are interdependent; consequently, the overall performance of the system is affected for each service invoked simultaneously during a high demand under the SOA pattern [75]. This pattern cannot be changed since all the services developed remain loosely coupled and any change requires the rebuilding or reimplementation of the entire application (coarse-grained services) [62,79–81]; therefore, the MSA variant is approximately 54.21% more efficient than the SOA variant in terms of total query numbers and average response times.
- API gateway: It is important to consider that SOA has a centralized governance of services, which also affects the deliveries and responses of queries to users [33]. We have seen how services develop without considering the weak points of this technology, e.g., not considering the centralization of data (without instances) or not considering a Cloud infrastructure for the operation of the services. The resulting increased traffic of the API of HTTP resources degrades the overall performance of the application until its collapse [111,112]. However, for MSA, applying decentralized and independent governance requires applying additional security measures unlike SOA (e.g., authenticating the user identity for each message sent and/or received through the signal mechanisms, the use of custom encryption PBKDF2 or the use of the ticket-based protocol CAS 3.0). These additional mechanisms are intended to establish a secure communication channel between the different microservices (internal and external) of computer attacks (e.g., man-in-the-middle, DoS or other) [104]. Consequently, it brings with it an increase in the general network traffic between the microservices and their infrastructure, demonstrating a significant increase in the average transfer of

data to the users; therefore, the SOA variant is approximately 73.80% more efficient than the MSA variant in terms of network consumption.

The SPIDEP-SOA-K8s-T1 and SPIDEP-MSA-K8s-T1 environments do not have a noticeable difference in the results compared to the K8s-T2 environments, but their trend is similar to the previous environment (approximately 9.31% and approximately 49.05% for each case) because this T1 environment executes multiple HTTP queries to a single service (service E), while the T2 environments execute multiple HTTP queries to two services simultaneously (services E and G).

Regarding the second finding, the four engine health reports of the K8s-T1/T2 environments meet the infrastructure performance (CPU and RAM) under a demand of 50 concurrent users (connections), since the CPU values are lower than 80% (8% to approximately 30%) and the memory levels are lower than 70% (10% to approximately 15%) [113,114]; however, the network I/O value of the SPIDEP-MSA-K8s-T1/T2 environments shows a significant consumption in the bandwidth traffic between the services and the Dockers instances in comparison with the SPIDEP-SOA-K8s-T1/T2 environments. This high consumption is because the MSA variant requires managing more specialized coordination to redirect queries from external platforms to internal microservices (or vice versa) within the Cloud environment [115]. In addition, it must actively route and validate user requests through the API gateway to the respective instances of the microservices [74].

In the third finding, to determine if the SPIDEP variants meet the software attribute (i.e., versatile high-performance), we have analyzed and contrasted the results obtained from the various stress tests of each platform. Considering the above, the MSA variant is the most appropriate for our needs, since it optimally meets the following three important aspects: (i) Scalability—MSA can be scaled individually when running a heavy workload by replicating the microservices on several containers and not replicating those that are underutilized (i.e., maximize the performance with minimal cost); therefore, microservices can handle the increase in demand without latency being significantly degraded, but this requires consuming a significant bandwidth to mitigate the latency [110,114,116]; (ii) Versatility—MSA allows adding and integrating new functionalities to the platform without affecting the availability of the other microservices. Therefore, each team has the autonomy to decide which is the best technology for the development of the service, according to the needs identified [7]; and (iii) Performance—a platform geared towards high demand requires that all its computing resources be distributed in the Cloud, which allows a shorter response time for the services; however, the platform becomes more demanding with the computational load of the instances.

7. Conclusions and Future Work

In this article, we have presented all the steps involved in the design, implementation and deployment of the SPIDEP platform and its RC variants based on the SOA and MSA architectural patterns; the platform is focused on remote telemonitoring of the elderly. Therefore, our purpose is to offer a replicable framework to support the early diagnosis of infectious diseases and their derivatives by using recent ICT developments through artificial intelligence algorithms (e.g., deep learning and machine learning-based inference systems), with the added value of reducing logistical costs for medical institutions.

Therefore, it was necessary to evaluate which of the SPIDEP variants are best suited for our versatile high-performance concept, considering the strengths and weaknesses of each architecture and how they behave in a high demand scenario. As a result, we have analyzed and contrasted the performance of each variant vs. the metrics obtained from the various performance tests (i.e., response time, efficient use of infrastructure and network consumption), which found that MSA is a better performer in terms of the performance quality attribute (approximately 54.21%). In the same manner, when processing multiple requests for various services, the response time was lower compared to SOA (approximately 7.34%), but the bandwidth consumption in MSA was more significant than SOA (approximately 73.80%).

Based on the ideas presented, we feel that the implementation of SOA and MSA depends on the capabilities and needs of organizations (e.g., performance, flexibility, availability, interoperability or other characteristics in return for known and unknown consequences) [33,63]. Therefore, within the eHealth context and based on the results obtained, we observe that MSA is capable of meeting the majority of needs in support of medical decision making and is adaptable to different types of clinical systems (e.g., EHR, IoHT or telemonitoring systems) and various infrastructure solutions (e.g., nursing homes, hospitals and public health management) [16,62,117]; however, this brings with it many challenges [51,74,94,99,105].

The SOA and MSA architectural patterns can be considered complementary allies for an interenterprise or interbusiness architecture that confers a suite of different services, rather than being competitors [33,118,119], i.e., combining the SOA and MSA attributes within an environment. Therefore, we plan to work on a proposal for an intergenerational ecosystem for SOA-MSA, with the aim of developing a basis for the integration and interconnection of the different eHealth applications involved in medical organizations in conjunction with microservices adapted to machine learning-based inference systems to perform specialized tasks in decision support in the prevention, monitoring and treatment of various conditions. Additionally, we are exploring the possibility of extending our framework to other eHealth areas (e.g., a patient monitoring system for hemodialysis) [120,121], early prediction of COVID-19 [122–125] or prediction of heart and kidney risks in diabetic patients [126–128]).

Another possible area for future research would be the adaptation of this proposal to other sectors of industry 4.0 (e.g., smart buildings or tourism) [8,129–132]; however, additional research is needed to obtain sufficient results to demonstrate the robustness of the architecture in terms of the adaptations of the attributes for these sectors. Our findings will be published in the near future.

Author Contributions: Conceptualization, J.M.G.-P. and M.V.-L.; methodology, H.C.-G., L.M.-P., J.M.G.-P., and M.V.-L.; software, H.C.-G., L.M.-P. and M.V.-L.; validation, H.C.-G., D.R.-P., G.S., M.-L.P.-L., and J.M.G.-P.; formal analysis, H.C.-G., J.M.G.-P. and M.-L.P.-L.; investigation, H.C.-G., L.M.-P., M.V.-L., J.M.G.-P., D.R.-P., G.S., and M.-L.P.-L.; resources, J.M.G.-P., M.-L.P.-L., G.S. and D.R.-P.; data curation, M.-L.P.-L. and G.S.; writing—original draft preparation, H.C.-G., L.M.-P., J.M.G.-P., and M.V.-L.; writing—review and editing, H.C.-G., L.M.-P., M.V.-L. and J.M.G.-P.; visualization, H.C.-G., and L.M.-P.; supervision, M.V.-L. and J.M.G.-P.; project administration, J.M.G.-P.; funding acquisition, J.M.G.-P. and M.V.-L. All authors have read and agreed to the published version of the manuscript.

Funding: This study forms part of the AC16/00061 project entitled "Design and implementation of a low-cost intelligent system for prediagnosis and telecare of infectious diseases in older people (SPIDEP)", which was financed by resources from the Instituto de Salud Carlos III of the Ministry of Science, Innovation of Spain, the Foundation for Biomedical Research of the Prince of Asturias University Hospital (FUHUPA), the National Secretariat of Science, Technology and Innovation of Panama (SENACYT-PANAMA) through the National Research System (SNI-PANAMA), and international scholarships (doctoral studies). In addition, this study forms part of the 2nd Joint Call for Research and Innovation (ERANet-LAC) of the Seventh Framework Program for Research and Technological Development (FP7) of the European Union under contract number ELAC2015/T09-0819 SPIDEP.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Exclude this statement if the study did not report any data.

Acknowledgments: We would like to extend special thanks to all the clinical and healthcare staff of the nursing homes for the elderly, the Francisco de Vitoria and Cisneros Centres of the Department of Social Policies and Family of the Community of Madrid in Spain and Nuestra Señora Carmen and Carls George in the Dominican Republic.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Van Eyk, E.; Toader, L.; Talluri, S.; Versluis, L.; Uta, A.; Iosup, A. Serverless is More: From PaaS to Present Cloud Computing. *IEEE Internet Comput.* **2018**, *22*, 8–17. [CrossRef]
2. Pfandzelter, T.; Bermbach, D. tinyFaaS: A Lightweight FaaS Platform for Edge Environments. In Proceedings of the 2020 IEEE International Conference on Fog Computing (ICFC), Sydney, Australia, 21–24 April 2020; pp. 17–24.
3. Gadepalli, P.K.; Peach, G.; Cherkasova, L.; Aitken, R.; Parmer, G. Challenges and Opportunities for Efficient Serverless Computing at the Edge. In Proceedings of the 2019 38th Symposium on Reliable Distributed Systems (SRDS), Lyon, France, 1–4 October 2019; pp. 261–2615.
4. Sewak, M.; Singh, S. Winning in the Era of Serverless Computing and Function as a Service. In Proceedings of the 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, India, 6–8 April 2018; pp. 1–5.
5. Kim, Y.; Cha, G. Design of the Cost Effective Execution Worker Scheduling Algorithm for FaaS Platform Using Two-Step Allocation and Dynamic Scaling. In Proceedings of the 2018 IEEE 8th International Symposium on Cloud and Service Computing (SC2), Paris, France, 18–21 November 2018; pp. 131–134. [CrossRef]
6. Khalique, F.; Khan, S.A.; Nosheen, I. A Framework for Public Health Monitoring, Analytics and Research. *IEEE Access* **2019**, *7*, 101309–101326. [CrossRef]
7. Suryotrisongko, H.; Jayanto, D.P.; Tjahyanto, A. Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot. *Procedia Comput. Sci.* **2017**, *124*, 736–743. [CrossRef]
8. Artemenko, O.; Pasichnyk, V.; Kunanec, N.; Tabachyshyn, D. Using context analysys for providing real time recommendations in e-tourism mobile location-based recommender systems. In Proceedings of the 2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 17–20 September 2019; Volume 3, pp. 166–169.
9. Tsai, Y.-T.; Fan, C.-L.; Lo, C.-L.; Huang, S.-H. SmartLohas: A Smart Assistive System for Elder People. In Proceedings of the 2017 14th International Symposium on Pervasive Systems, Algorithms and Networks & 2017 11th International Conference on Frontier of Computer Science and Technology & 2017 Third International Symposium of Creative Computing (ISPAN-FCST-ISCC, Exeter, UK, 21–23 June 2017; pp. 369–374.
10. Mendes, D.; Jorge, D.; Pires, G.; Panda, R.; Antonio, R.; Dias, P.; Oliveira, L. VITASENIOR-MT: A distributed and scalable cloud-based telehealth solution. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; pp. 767–772. [CrossRef]
11. Kontis, V.; Bennett, J.E.; Mathers, C.D.; Li, G.; Foreman, K.; Ezzati, M. Future life expectancy in 35 industrialised countries: Projections with a Bayesian model ensemble. *Lancet* **2017**, *389*, 1323–1335. [CrossRef]
12. Wang, Z.; Saho, K.; Tomiyama, H.; Meng, L. Gender Classification of Elderly People using Doppler Radar Images based on Machine Learning. In Proceedings of the 2019 International Conference on Advanced Mechatronic Systems (ICAMechS), Kusatsu, Japan, 26–28 August 2019; pp. 305–310.
13. Rizvi, S.; Sohail, I.; Saleem, M.M.; Irtaza, A.; Zafar, M.; Syed, M. A Smart Home Appliances Power Management System for Handicapped, Elder and Blind People. In Proceedings of the 2018 4th International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 13–14 August 2018; pp. 1–4.
14. Kulik, C.T.; Ryan, S.; Harper, S.; George, G. Aging Populations and Management. *Acad. Manag. J.* **2014**, *57*, 929–935. [CrossRef]
15. Solé-Casals, M.; Chirveches-Pérez, E.; Puigoriol-Juvanteny, E.; Nubó-Puntí, N.; Chabrera-Sanz, C.; Subirana-Casacuberta, M. Perfil y resultados del paciente frágil valorado por la Enfermera de Práctica Avanzada en un servicio de urgencias. *Enfermería Clínica* **2018**, *28*, 365–374. [CrossRef] [PubMed]
16. Ianculescu, M.; Alexandru, A.; Neagu, G.; Pop, F. Microservice-Based Approach to Enforce an IoHT Oriented Architecture. In Proceedings of the 2019 E-Health and Bioengineering Conference (EHB), Iasi, Romania, 21–23 November 2019; pp. 1–4.
17. Jarwar, M.A.; Ali, S.; Chong, I. Exploring Web Objects enabled Data-Driven Microservices for E-Health Service Provision in IoT Environment. In Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 17–19 October 2018; pp. 112–117.
18. Kocak, S.; Artug, T.; Tulum, G. A Preliminary Study for Remote Healthcare System: Activity Classification for Elder People with on Body Sensors. In Proceedings of the 2018 6th International Conference on Control Engineering & Information Technology (CEIT), Istanbul, Turkey, 25–27 October 2018; pp. 1–3.
19. Gaddam, A.; Mukhopadhyay, S.C.; Gupta, G.S. Necessity of a bed-sensor in a smart digital home to care for elder-people. In Proceedings of the 2008 IEEE Sensors, Lecce, Italy, 26–29 October 2008; pp. 1340–1343.
20. Wishner, L.; Sollevad, J.; Rudowitz, P.; Paradise, R.; Antonisse, J. A Look at Rural Hospital Closures and Implications for Access to Care: Three Case Studies. 2016. Available online: https://www.kff.org/report-section/a-look-at-rural-hospital-closures-and-implications-for-access-to-care-three-case-studies-issue-brief/ (accessed on 12 January 2021).
21. Calderon-Gomez, H.; Mendoza-Pitti, L.; Vargas-Lombardo, M.; Gomez-Pulido, J.M.; Castillo-Sequera, J.L.; Sanz-Moreno, J.; Sencion, G. Telemonitoring System for Infectious Disease Prediction in Elderly People Based on a Novel Microservice Architecture. *IEEE Access* **2020**, *8*, 118340–118354. [CrossRef]
22. Sanz-Moreno, J.; Gómez-Pulido, J.; Garcés, A.; Calderón-Gómez, H.; Vargas-Lombardo, M.; Castillo-Sequera, J.L.; Luque, M.L.P.; Toro, R.; Sención-Martínez, G. mHealth System for the Early Detection of Infectious Diseases Using Biomedical Signals. In *The Importance of New Technologies and Entrepreneurship in Business Development: In The Context of Economic Diversity in Developing Countries*; Metzler, J.B., Ed.; Springer: Berlin/Heidelberg, Germany, 2020; pp. 203–213.

23. Calderon-Gomez, H.; Garces-Jimenez, A.; Vargas-Lombardo, M.; Gomez-Pulido, J.M.; Polo-Luque, M.-L.; Castillo, J.; Sencion, G.; Moreno, J.S. Proposal Using the Cloud Architecture in System for the Early Detection of Infectious Diseases in Elderly People Fed by Biosensors Records. In Proceedings of the 2019 7th International Engineering, Sciences and Technology Conference (IESTEC), Panama, Panama, 9–11 October 2019; pp. 631–634.
24. Calderón-Gómez, H.; Navarro-Marín, F.; Gómez-Pulido, J.M.; Castillo-Sequera, J.L.; Garcés-Jiménez, A.; Polo-Luque, M.-L.; Sanz-Moreno, J.; Sención, G.; Vargas-Lombardo, M. Development of ehealth applications-based on microservices in a cloud architecture. *Rev. Iber. Sist. Tecnol. Inf.* **2019**, *2019*, 81–93.
25. Baldominos, A.; Ogul, H.; Colomo-Palacios, R.; Sanz-Moreno, J.; Gómez-Pulido, J.M. Infection prediction using physiological and social data in social environments. *Inf. Process. Manag.* **2020**, *57*, 102213. [CrossRef]
26. Aazam, M.; Zeadally, S.; Harras, K.A. Deploying Fog Computing in Industrial Internet of Things and Industry 4.0. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4674–4682. [CrossRef]
27. Patel, A.R.; Azadi, S.; Babaee, M.H.; Mollaei, N.; Patel, K.L.; Mehta, D.R. Significance of Robotics in Manufacturing, Energy, Goods and Transport Sector in Internet of Things (IoT) Paradigm. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 16–18 August 2018; pp. 1–4.
28. Kousiouris, G.; Tsarsitalidis, S.; Psomakelis, E.; Koloniaris, S.; Bardaki, C.; Tserpes, K.; Nikolaidou, M.; Anagnostopoulos, D. A microservice-based framework for integrating IoT management platforms, semantic and AI services for supply chain management. *ICT Express* **2019**, *5*, 141–145. [CrossRef]
29. Baresi, L.; Mendonca, D.F. Towards a Serverless Platform for Edge Computing. In Proceedings of the 2019 IEEE International Conference on Fog Computing (ICFC), Prague, Czech Republic, 24–26 June 2019; pp. 1–10.
30. Da Silva, M.A.P.; Times, V.C.; De Araujo, A.M.C.; Da Silva, P.C. A Microservice-Based Approach for Increasing Software Reusability in Health Applications. In Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019; pp. 1–8.
31. Castrillón-Betancur, J.C.; Flórez-Arango, J.F. Archetypes, terminologies and semantic interoperability in health. Arquetipos, terminologías e interoperabilidad semántica en salud. *Rev. Cuba. Investig. Biomed.* **2015**, *34*, 365–377. Available online: http://ref.scielo.org/kp9gdp (accessed on 14 January 2021).
32. Chaim, R.M.; Oliveira, E.C.; Araujo, A.P.F. Technical specifications of a service-oriented architecture for semantic interoperability of HER—Electronic health records. In Proceedings of the 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), Lisbon, Portugal, 21–24 June 2017; pp. 1–6.
33. Andriyanto, A.; Doss, R.; Yustianto, P. Adopting SOA and Microservices for Inter-enterprise Architecture in SME Communities. In Proceedings of the 2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE), Denpasar, Indonesia, 3–4 October 2019; Volume 6, pp. 282–287.
34. Hutapea, R.C.A.; Wahyudi, A.P. Design Quality Measurement for Service Oriented Software on Service Computing System: A Systematic Literature Review. In Proceedings of the 2018 International Conference on Information Technology Systems and Innovation (ICITSI), Bandung, Indonesia, 22–26 October 2018; pp. 375–380.
35. Bouguettaya, A.; Singh, M.; Huhns, M.; Sheng, Q.Z.; Dong, H.; Yu, Q.; Neiat, A.G.; Mistry, S.; Benatallah, B.; Medjahed, B.; et al. A service computing manifesto. *Commun. ACM* **2017**, *60*, 64–72. [CrossRef]
36. Mandal, A.K.; Sarkar, A. Service Oriented System design: Domain Specific Model based approach. In Proceedings of the 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 15–17 August 2016; pp. 489–494.
37. Gunawan, L.K.; Tama, N.S.; Nurjannah, R.; Fauziyah, U.; Fajar, A.N.; Qomariyah, N.N. WELTY System Design Based on Service Oriented Architecture for Smart HealthCare. In Proceedings of the 2019 International Conference on ICT for Smart Society (ICISS), Bandung, Indonesia, 19–20 November 2019; Volume 7, pp. 1–4.
38. Gavrilov, G.; Vlahu-Gjorgievska, E.; Trajkovik, V. Healthcare data warehouse system supporting cross-border interoperability. *Health Inform. J.* **2020**, *26*, 1321–1332. [CrossRef]
39. Amin, M.M.; Sutrisman, A.; Stiawan, D.; Ermatita, E.; Alzahrani, M.Y.; Budiarto, R. Interoperability framework for integrated e-health services. *Bull. Electr. Eng. Inform.* **2020**, *9*, 354–361. [CrossRef]
40. Chronaki, C.; Estelrich, A.; Cangioli, G.; Melgara, M.; Kalra, D.; Gonzaga, Z.; Garber, L.; Blechman, E.; Ferguson, J.; Kay, S. Interoperability standards enabling cross-border patient summary exchange. *Stud. Heal. Technol. Inform.* **2014**, *205*, 256–260.
41. European Commission. eHealth Action Plan 2012–2020—Innovative Healthcare for the 21st Century. 2012. Available online: https://ec.europa.eu/health//sites/health/files/ehealth/docs/com_2012_736_en.pdf (accessed on 16 January 2021).
42. Chang, S.H. A Systematic Analysis and Design Approach to Develop Adaptable Services in Service Oriented Computing. In Proceedings of the 2007 IEEE Congress on Services (Services 2007), Salt Lake City, UT, USA, 9–13 July 2007; pp. 375–378.
43. Park, J.; Moon, M.; Yeom, K. The BCD View Model: Business Analysis View, Service Composition View and Service Design View for Service Oriented Software Design and Development. In Proceedings of the 2008 12th IEEE International Workshop on Future Trends of Distributed Computing Systems, Kunming, China, 21–23 October 2008; pp. 37–43. [CrossRef]
44. Kufner, J.; Marik, R. Restful State Machines and SQL Database. *IEEE Access* **2019**, *7*, 144603–144617. [CrossRef]
45. Hameed, R.T.; Mohamad, O.A.; Hamid, O.T.; Tapus, N. Patient monitoring system based on e-health sensors and web services. In Proceedings of the 2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Ploiesti, Romania, 30 June–2 July 2016; pp. 1–6.

46. Srinivasan, C.; Charan, G.; Babu, P.C.S. An IoT based SMART patient health monitoring system. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *18*, 1657–1664. [CrossRef]
47. Peng, C.; Goswami, P.; Bai, G. Linking Health Web Services as Resource Graph by Semantic REST Resource Tagging. *Procedia Comput. Sci.* **2018**, *141*, 319–326. [CrossRef]
48. Silva, B.M.C.; Rodrigues, J.J.P.C.; Ramos, A.; Saleem, K.; De La Torre, I.; Rabelo, R.L. A Mobile Health System to Empower Healthcare Services in Remote Regions. In Proceedings of the 2019 IEEE International Conference on E-health Networking, Application & Services (HealthCom), Bogota, Colombia, 14–16 October 2019; pp. 1–6.
49. Vayghan, L.A.; Saied, M.A.; Toeroe, M.; Khendek, F. Microservice Based Architecture: Towards High-Availability for Stateful Applications with Kubernetes. In Proceedings of the 2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS), Sofia, Bulgaria, 22–26 July 2019; pp. 176–185.
50. Cojocaru, M.-D.; Uta, A.; Oprescu, A.-M. Attributes Assessing the Quality of Microservices Automatically Decomposed from Monolithic Applications. In Proceedings of the 2019 18th International Symposium on Parallel and Distributed Computing (ISPDC), Amsterdam, The Netherlands, 3–7 June 2019; pp. 84–93.
51. Hassan, S.; Bahsoon, R.; Kazman, R. Microservice transition and its granularity problem: A systematic mapping study. *Softw. Pr. Exp.* **2020**, *50*, 1651–1681. [CrossRef]
52. De Lauretis, L. From Monolithic Architecture to Microservices Architecture. In Proceedings of the 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Berlin, Germany, 27–30 October 2019; pp. 93–96. [CrossRef]
53. Guaman, D.; Yaguachi, L.; Samanta, C.C.; Danilo, J.H.; Soto, F. Performance evaluation in the migration process from a monolithic application to microservices. In Proceedings of the 2018 13th Iberian Conference on Information Systems and Technologies (CISTI), Caceres, Spain, 13–16 June 2018; pp. 1–8.
54. Akbulut, A.; Perros, H.G. Performance Analysis of Microservice Design Patterns. *IEEE Internet Comput.* **2019**, *23*, 19–27. [CrossRef]
55. Di Francesco, P.; Lago, P.; Malavolta, I. Architecting with microservices: A systematic mapping study. *J. Syst. Softw.* **2019**, *150*, 77–97. [CrossRef]
56. Li, S.; Zhang, H.; Jia, Z.; Li, Z.; Zhang, C.; Li, J.; Gao, Q.; Ge, J.; Shan, Z. A dataflow-driven approach to identifying microservices from monolithic applications. *J. Syst. Softw.* **2019**, *157*, 110380. [CrossRef]
57. Carranza-García, F.; García-Moreno, F.M.; Rodriguez-Dominguez, C.; Garrido, J.L.; Edo, M.B.; Rodriguez-Fortiz, M.J.; Pérez-Mármol, J.M. *Supporting Active Ageing Interventions with Web and Mobile/Wearable Technologies and Using Microservice Oriented Architectures*; Metzler, J.B., Ed.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 1016, pp. 114–123.
58. Garcia-Moreno, F.M.; Bermudez-Edo, M.; Garrido, J.L.; Rodríguez-García, E.; Pérez-Mármol, J.M.; Rodríguez-Fórtiz, M.J. A Microservices e-Health System for Ecological Frailty Assessment using Wearables. *Sensors* **2020**, *20*, 3427. [CrossRef]
59. Sianaki, O.A.; Yousefi, A.; Tabesh, A.R.; Mahdavi, M. Internet of Everything and Machine Learning Applications: Issues and Challenges. In Proceedings of the 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), Krakow, Poland, 16–18 May 2018; pp. 704–708.
60. Houmani, Z.; Balouek-Thomert, D.; Caron, E.; Parashar, M. Enhancing microservices architectures using data-driven service discovery and QoS guarantees. In Proceedings of the 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), Melbourne, Australia, 11–14 May 2020; pp. 290–299.
61. Frade, S.; Freire, S.M.; Sundvall, E.; Patriarca-Almeida, J.H.; Cruz-Correia, R. Survey of openEHR storage implementations. In Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems, Porto, Portugal, 20–22 June 2013; pp. 303–307.
62. Rademacher, F.; Sachweh, S.; Zundorf, A. Differences between Model-Driven Development of Service-Oriented and Microservice Architecture. In Proceedings of the 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, Sweden, 5–7 April 2017; pp. 38–45.
63. Firmansyah, T.; Ganeswangga, A.; Fajar, A.N. Transforming the Monolith E-Procurement Application. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *9*, 4780–4784. [CrossRef]
64. Kazanavicius, J.; Mazeika, D. Migrating Legacy Software to Microservices Architecture. In Proceedings of the 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 25 April 2019; pp. 1–5.
65. Zhang, Y.; Liu, B.; Dai, L.; Chen, K.; Cao, X. Automated Microservice Identification in Legacy Systems with Functional and Non-Functional Metrics. In Proceedings of the 2020 IEEE International Conference on Software Architecture (ICSA), Salvador, Brazil, 16–20 March 2020; pp. 135–145.
66. Selmadji, A.; Seriai, A.-D.; Bouziane, H.L.; Mahamane, R.O.; Zaragoza, P.; Dony, C. From Monolithic Architecture Style to Microservice one Based on a Semi-Automatic Approach. In Proceedings of the 2020 IEEE International Conference on Software Architecture (ICSA), Salvador, Brazil, 16–20 March 2020; pp. 157–168.
67. Fundación para la Investigación Biomédica del Hospital Universitario Príncipe de Asturias. Design and Implementation of a Low Cost Smart System for Pre-Diagnosis and Telecare of Infectious Diseases in Elderly People | EU-CELAC. 2017. Available online: https://www.eucelac-platform.eu/project/design-and-implementation-low-cost-smart-system-pre-diagnosis-and-telecare-infectious (accessed on 11 February 2021).
68. Catarinucci, L.; De Donno, D.; Mainetti, L.; Palano, L.; Patrono, L.; Stefanizzi, M.L.; Tarricone, L. An IoT-Aware Architecture for Smart Healthcare Systems. *IEEE Internet Things J.* **2015**, *2*, 515–526. [CrossRef]

69. Javed, A.; Robert, J.; Heljanko, K.; Främling, K. IoTEF: A Federated Edge-Cloud Architecture for Fault-Tolerant IoT Applications. *J. Grid Comput.* **2020**, *18*, 57–80. [CrossRef]
70. Di Francesco, P. Architecting Microservices. In Proceedings of the 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, Sweden, 5–7 April 2017; pp. 224–229.
71. Zhou, S. Improving Collaboration Efficiency in Fork-Based Development. In Proceedings of the 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), San Diego, CA, USA, 11–15 November 2019; pp. 1218–1221.
72. Duszynski, S.; Tenev, V.L.; Becker, M. N-way Diff: Set-based Comparison of Software Variants. In Proceedings of the 2020 Working Conference on Software Visualization (VISSOFT), Adelaide, Australia, 28 September–2 October 2020; pp. 72–83.
73. Chua, B. Detecting sustainable programming languages through forking on open source projects for survivability. In Proceedings of the 2015 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Gaithersburg, MD, USA, 2–5 November 2015; pp. 120–124. [CrossRef]
74. De Sanctis, M.; Muccini, H.; Vaidhyanathan, K. Data-driven Adaptation in Microservice-based IoT Architectures. In Proceedings of the 2020 IEEE International Conference on Software Architecture (ICSA), Salvador, Brazil, 16–20 March 2020; pp. 59–62.
75. Papazoglou, M.P.; Heuvel, W.-J.V.D. Service oriented architectures: Approaches, technologies and research issues. *VLDB J.* **2007**, *16*, 389–415. [CrossRef]
76. Zhao, H.; Chen, X.; Zhang, W.; Liang, P.; Wang, J.; Huang, W. SOA Patterns Selection and Application Based on Software Quality Requirements. In Proceedings of the 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 23–25 November 2018; pp. 361–364.
77. Rahim, R.; Suhardi; Kurniawan, N.B. Designing statistical metadata service computing system based on service oriented Architecture (SOA). In Proceedings of the 2017 International Conference on Information Technology Systems and Innovation (ICITSI), Bandung, Indonesia, 23–24 October 2017; pp. 120–127.
78. Pulparambil, S.; Baghdadi, Y. A Comparison Framework for SOA Maturity Models. In Proceedings of the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015; pp. 1102–1107.
79. Cheng, B.; Zhu, D.; Zhao, S.; Chen, J. Situation-Aware IoT Service Coordination Using the Event-Driven SOA Paradigm. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 349–361. [CrossRef]
80. Galster, M.; Avgeriou, P. Qualitative Analysis of the Impact of SOA Patterns on Quality Attributes. In Proceedings of the 2012 12th International Conference on Quality Software, Xi'an, China, 27–29 August 2012; pp. 167–170.
81. Pulparambil, S.; Baghdadi, Y. Towards a model for soa adoption based on methodical aspects. In Proceedings of the 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 18–19 July 2017; pp. 105–110.
82. Rodríguez, G.; Teyseyre, A.; Soria, Á.; Berdun, L. A visualization tool to detect refactoring opportunities in SOA applications. In Proceedings of the 2017 XLIII Latin American Computer Conference (CLEI), Cordoba, Argentina, 4–8 September 2017; pp. 1–10.
83. Rida, B.; Ahmed, E. Multiview SOA: Extending SOA using a private cloud computing as SaaS. In Proceedings of the 2015 International Conference on Cloud Technologies and Applications (CloudTech), Marrakech, Morocco, 2–4 June 2015; pp. 1–5.
84. Zhang, H.; Yang, X. Cloud Computing Architecture Based-On SOA. In Proceedings of the 2012 Fifth International Symposium on Computational Intelligence and Design, Hangzhou, China, 28–29 October 2012; pp. 369–373. [CrossRef]
85. Omar, W.; Taleb-Bendiab, A. E-health support services based on service-oriented architecture. *IT Prof.* **2006**, *8*, 35–41. [CrossRef]
86. Appandairaj, A.; Murugappan, S. Service Oriented Architecture Design for Web Based Home Banking Systems with Cloud Based Service. *Int. J. Emerg. Technol. Adv. Eng.* **2013**, *3*, 138–143.
87. Khoonsari, P.E.; Moreno, P.; Bergmann, S.; Burman, J.; Capuccini, M.; Carone, M.; Cascante, M.; De Atauri, P.; Foguet, C.; Gonzalez-Beltran, A.N.; et al. Interoperable and scalable data analysis with microservices: Applications in metabolomics. *Bioinformatics* **2019**, *35*, 3752–3760. [CrossRef]
88. Wan, X.; Guan, X.; Wang, T.; Bai, G.; Choi, B.-Y. Application deployment using Microservice and Docker containers: Framework and optimization. *J. Netw. Comput. Appl.* **2018**, *119*, 97–109. [CrossRef]
89. Gan, Y.; Delimitrou, C. The Architectural Implications of Cloud Microservices. *IEEE Comput. Arch. Lett.* **2018**, *17*, 155–158. [CrossRef]
90. Shamim, S.I.; Bhuiyan, F.A.; Rahman, A. XI Commandments of Kubernetes Security: A Systematization of Knowledge Related to Kubernetes Security Practices. In Proceedings of the 2020 IEEE Secure Development (SecDev), Atlanta, GA, USA, 28–30 September 2020; pp. 58–64.
91. He, Z. Novel Container Cloud Elastic Scaling Strategy based on Kubernetes. In Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 12–14 June 2020; pp. 1400–1404.
92. Baldominos, A.; Ogul, H.; Colomo-Palacios, R. Infection Diagnosis using Biomedical Signals in Small Data Scenarios. In Proceedings of the 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS), Cordoba, Spain, 5–7 June 2019; pp. 38–43.
93. O'Brien, O.; O'Reilly, R.D. Beats-Per-Minute (BPM): A Microservice-based Platform for the Monitoring of Health Related Data via Activity Trackers. In Proceedings of the 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom), Ostrava, Czech Republic, 17–20 September 2018; pp. 1–7.
94. Gong, Y.; Gu, F.; Chen, K.; Wang, F. The Architecture of Micro-services and the Separation of Frond-end and Back-end Applied in a Campus Information System. In Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China, 25–27 August 2020; pp. 321–324.

95. Munonye, K.; Martinek, P. Evaluation of Data Storage Patterns in Microservices Archicture. In Proceedings of the 2020 IEEE 15th International Conference of System of Systems Engineering (SoSE), Budapest, Hungary, 2–4 June 2020; pp. 373–380.
96. Bao, L.; Wu, C.; Bu, X.; Ren, N.; Shen, M. Performance Modeling and Workflow Scheduling of Microservice-Based Applications in Clouds. *IEEE Trans. Parallel Distrib. Syst.* **2019**, *30*, 2114–2129. [CrossRef]
97. Akbulut, A.; Perros, H.G. Software Versioning with Microservices through the API Gateway Design Pattern. In Proceedings of the 2019 9th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, 5–7 June 2019; pp. 289–292.
98. Tapia, F.; Ángel Mora, M.; Fuertes, W.; Aules, H.; Flores, E.; Toulkeridis, T. From Monolithic Systems to Microservices: A Comparative Study of Performance. *Appl. Sci.* **2020**, *10*, 5797. [CrossRef]
99. Götz, B.; Schel, D.; Bauer, D.; Henkel, C.; Einberger, P.; Bauernhansl, T. Challenges of Production Microservices. *Procedia CIRP* **2018**, *67*, 167–172. [CrossRef]
100. Rademacher, F.; Sachweh, S.; Zundorf, A. Aspect-Oriented Modeling of Technology Heterogeneity in Microservice Architecture. In Proceedings of the 2019 IEEE International Conference on Software Architecture (ICSA), Hamburg, Germany, 25–29 March 2019; pp. 21–30.
101. Álvaro, B.; Solé, M.; Huélamo, A.; Solans, D.; Pérez, M.S.; Muntés-Mulero, V. Graph-based root cause analysis for service-oriented and microservice architectures. *J. Syst. Softw.* **2020**, *159*, 110432. [CrossRef]
102. Chen, L. Microservices: Architecting for Continuous Delivery and DevOps. In Proceedings of the 2018 IEEE International Conference on Software Architecture (ICSA), Seattle, WA, USA, 30 April–4 May 2018; pp. 39–397.
103. Balalaie, A.; Heydarnoori, A.; Jamshidi, P. Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture. *IEEE Softw.* **2016**, *33*, 42–52. [CrossRef]
104. Roca, S.; Sancho, J.; García, J.; Álvaro, A. Microservice chatbot architecture for chronic patient support. *J. Biomed. Inform.* **2020**, *102*, 103305. [CrossRef]
105. Gribaudo, M.; Iacono, M.; Manini, D. Performance Evaluation of Replication Policies in Microservice Based Architectures. *Electron. Notes. Comput. Sci.* **2018**, *337*, 45–65. [CrossRef]
106. Lonetti, F.; Marchetti, E. *Chapter Three—Emerging Software Testing Technologies*; Memon, C., Ed.; Elsevier: Amsterdam, The Netherlands, 2018; pp. 91–143.
107. Shoumik, F.S.; Talukder, I.M.M.; Jami, A.I.; Protik, N.W.; Hoque, M. Scalable micro-service based approach to FHIR server with golang and No-SQL. In Proceedings of the 2017 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 22–24 December 2017; pp. 1–6.
108. Khan, R.; Amjad, M. Performance testing (load) of web applications based on test case management. *Perspect. Sci.* **2016**, *8*, 355–357. [CrossRef]
109. Ramakrishnan, R.; Kaur, A. Little's law based validation framework for load testing. *Inf. Softw. Technol.* **2019**, *108*, 88–98. [CrossRef]
110. Abdullah, M.; Iqbal, W.; Erradi, A. Unsupervised learning approach for web application auto-decomposition into microservices. *J. Syst. Softw.* **2019**, *151*, 243–257. [CrossRef]
111. Kumar, N.M.; Mallick, P.K. The Internet of Things: Insights into the building blocks, component interactions, and architecture layers. *Procedia Comput. Sci.* **2018**, *132*, 109–117. [CrossRef]
112. Alharbi, F.; Atkins, A.; Stanier, C. *Cloud Computing Adoption in Healthcare Organisations: A Qualitative Study in Saudi Arabia BT—Transactions on Large-Scale Data- and Knowledge-Centered Systems XXXV*; Hameurlain, A., Küng, J., Wagner, R., Sakr, S., Razzak, I., Riyad, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; pp. 96–131.
113. Cortavitarte, J. Engine Health Report. 2021. Available online: https://guide.blazemeter.com/hc/en-us/articles/360000290058-Engine-Health-Report-Engine-Health-Report (accessed on 17 February 2021).
114. Khaleq, A.A.; Ra, I. Agnostic Approach for Microservices Autoscaling in Cloud Applications. In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 5–7 December 2019; pp. 1411–1415.
115. Hassan, S.; Ali, N.; Bahsoon, R. Microservice Ambients: An Architectural Meta-Modelling Approach for Microservice Granularity. In Proceedings of the 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg, Sweden, 3–7 April 2017; pp. 1–10.
116. Kiss, T.; DesLauriers, J.; Gesmier, G.; Terstyanszky, G.; Pierantoni, G.; Abu Oun, O.; Taylor, S.J.; Anagnostou, A.; Kovacs, J. A cloud-agnostic queuing system to support the implementation of deadline-based application execution policies. *Futur. Gener. Comput. Syst.* **2019**, *101*, 99–111. [CrossRef]
117. Enler, E.; Pentek, I.; Adamko, A. Healthcare Framework for Smarter Cities with bio-sensory data. In Proceedings of the 2020 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Mariehamn, Finland, 23–25 September 2020; pp. 337–342.
118. Xiao, Z.; Wijegunaratne, I.; Qiang, X. Reflections on SOA and Microservices. In Proceedings of the 2016 4th International Conference on Enterprise Systems (ES), Melbourne, Australia, 2–3 November 2016; pp. 60–67.
119. Sriraman, A.; Wenisch, T.F. μ Suite: A Benchmark Suite for Microservices. In Proceedings of the 2018 IEEE International Symposium on Workload Characterization (IISWC), Raleigh, NC, USA, 30 September–2 October 2018; pp. 1–12. [CrossRef]

120. Zainol, M.F.; Farook, R.S.M.; Hassan, R.; Halim, A.H.A.; Rejab, M.R.A.; Husin, Z. A New IoT Patient Monitoring System for Hemodialysis Treatment. In Proceedings of the 2019 IEEE Conference on Open Systems (ICOS), Pulau Pinang, Malaysia, 19–21 November 2019; pp. 46–50.
121. Takeda, Y.; Yokoyama, D.; Nakamichi, N.; Inaba, R.; Watanabe, K.; Yamada, T. Visualization of Remote Touch Panel for Dialysis Patient on Prototype Bed. In Proceedings of the 2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech), Kyoto, Japan, 10–12 March 2020; pp. 54–58.
122. Zhong, L.; Mu, L.; Li, J.; Wang, J.; Yin, Z.; Liu, D. Early Prediction of the 2019 Novel Coronavirus Outbreak in the Mainland China Based on Simple Mathematical Model. *IEEE Access* **2020**, *8*, 51761–51769. [CrossRef]
123. Rimsan, M.; Mahmood, A.K.; Umair, M.; Hassan, F. COVID-19: A Novel Framework to Globally Track Coronavirus Infected Patients using Blockchain. In Proceedings of the 2020 International Conference on Computational Intelligence (ICCI), Bandar Seri Iskandar, Malaysia, 8–9 October 2020; pp. 70–74.
124. Han, R.; Liu, Z.; Chen, C.L.P.; Xu, L.; Peng, G. Mortality prediction for COVID-19 patients via Broad Learning System. In Proceedings of the 2020 7th International Conference on Information, Cybernetics and Computational Social Systems (ICCSS), Guangzhou, China, 13–15 November 2020; pp. 837–842.
125. Casiraghi, E.; Malchiodi, D.; Trucco, G.; Frasca, M.; Cappelletti, L.; Fontana, T.; Esposito, A.A.; Avola, E.; Jachetti, A.; Reese, J.; et al. Explainable Machine Learning for Early Assessment of COVID-19 Risk Prediction in Emergency Departments. *IEEE Access* **2020**, *8*, 196299–196325. [CrossRef]
126. Ananthi, S.; Bhuvaneswari, V. Prediction of heart and kidney risks in diabetic prone population using fuzzy classification. In Proceedings of the 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 5–7 January 2017; pp. 1–6.
127. Patil, S.S.; Malpe, K. Implementation of Diabetic Retinopathy Prediction System using Data Mining. In Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 27–29 March 2019; pp. 1206–1210.
128. Qiao, L.; Zhu, Y.; Zhou, H. Diabetic Retinopathy Detection Using Prognosis of Microaneurysm and Early Diagnosis System for Non-Proliferative Diabetic Retinopathy Based on Deep Learning Algorithms. *IEEE Access* **2020**, *8*, 104292–104302. [CrossRef]
129. Saputra, R.Y.; Nugroho, L.E.; Kusumawardani, S.S. Collecting the Tourism Contextual Information data to support the tourism recommendation system. In Proceedings of the 2019 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 24–25 July 2019; pp. 79–84.
130. Bigheti, J.A.; Fernandes, M.M.; Godoy, E.D.P. Control as a Service: A Microservice Approach to Industry 4.0. In Proceedings of the 2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT), Naples, Italy, 4–6 June 2019; pp. 438–443. [CrossRef]
131. Prasetyo, Y.A.; Suhardi. Microservice Platform for Smart City: Concepts, Services and Technology. In Proceedings of the 2018 International Conference on Information Technology Systems and Innovation (ICITSI), Bandung, Indonesia, 22–26 October 2018; pp. 358–363.
132. Mendoza-Pitti, L.; Calderon-Gomez, H.; Vargas-Lombardo, M.; Gomez-Pulido, J.M.; Castillo-Sequera, J.L. Towards a service-oriented architecture for the energy efficiency of buildings: A systematic review. *IEEE Access* **2021**, *9*, 1. [CrossRef]

3 Otros Méritos

Igualmente, se presentaron otros resultados que han ayudado en el desarrollo de esta investigación científica, cuyos resultados avalan el impacto de esta tesis doctoral con diversas contribuciones hacia la comunidad científica.

3.1 Congresos Internacionales

3.1.1 LACAR 2019: Latin American Congress on Automation and Robotics 2019

- Título del manuscrito: "mHealth System for the Early Detection of Infectious Diseases Using Biomedical Signals"
- Tipo de manuscrito: Capítulo de libro - Springer
- Lugar: Cali, Colombia
- Fecha: 30 de octubre al 01 de noviembre de 2019
- DOI: https://doi.org/10.1007/978-3-030-40309-6_20
- https://doi.org/10.1007/978-3-030-40309-6_20
- Indexación: Scopus - SJR
- Referencia: Sanz-Moreno, J. et al. (2020). mHealth System for the Early Detection of Infectious Diseases Using Biomedical Signals. In: Martínez, A., Moreno, H., Carrera, I., Campos, A., Baca, J. (eds) Advances in Automation and Robotics Research. LACAR 2019. Lecture Notes in Networks and Systems, vol 112. Springer, Cham.

3.1.2 CSEI 2019: I Conference on computer science, electronics, and industrial engineering

- Título del manuscrito: "Desarrollo de aplicaciones eHealth basadas en microservicios en una arquitectura de Cloud"
- Tipo de manuscrito: Revista Indexada - Revista Ibérica de Sistemas e Tecnologias de Informação (Volume 2019, Issue E23, October 2019, Pages 81-93) – Edición no regular
- Lugar: Ambato, Ecuador
- Fecha: 28 al 31 de octubre de 2019
- URL: https://www.proquest.com/openview/d13bdc159901f62306f1d7f4fa9349f5
- Indexación: Scopus - SJR
- Referencia: Calderón-Gómez, H., Navarro-Marín, F., Gómez-Pulido, J. M., Castillo-Sequera, J. L., Garcés-Jiménez, A., Polo-Luque, M. L., ... & Vargas-Lombardo, M. (2019). Development of ehealth applications-based on microservices in a cloud architecture. RISTI Rev. Iber. Sist. Tecnol. Inf, 2019(E23), 81-93.

3.1.3 2019 7th International Engineering, Sciences and Technology Conference (IESTEC)

- Título del manuscrito: “Proposal using the cloud architecture in system for the early detection of infectious diseases in elderly people fed by biosensors records”
- Tipo de manuscrito: Conferences
- Editorial: IEEE
- Lugar: Panamá, Panamá
- Fecha: 9 al 11 de octubre de 2019
- URL: https://doi.org/10.1109/IESTEC46403.2019.00118
- Indexación: Scopus - SJR
- Referencia: H. Calderón-Gómez et al., "Proposal Using the Cloud Architecture in System for the Early Detection of Infectious Diseases in Elderly People Fed by Biosensors Records," 2019 7th International Engineering, Sciences and Technology Conference (IESTEC), 2019, pp. 631-634, doi: 10.1109/IESTEC46403.2019.00118.

3.2 Colaboración con proyectos de investigación

3.2.1 SPIDEP

El Proyecto SPIDEP (Diseño e implementación de un sistema inteligente de bajo coste para el prediagnóstico y la teleasistencia de enfermedades infecciosas en personas de edad avanzada) tiene como objetivo de construir un marco de desarrollo basado en las TICs para apoyar el diagnóstico precoz de enfermedades infecciosas en personas mayores. Por ello, este proyecto ha formado un consorcio internacional entre investigadores, técnicos y personal médico (Universidad de Alcalá, Área de Enfermedades Crónicas y Cáncer del Instituto Ramón y Cajal de Investigación Sanitaria – Madrid, España, Universidad Tecnológica de Panamá, Facultad de Medicina de la Universidad Autónoma de Santo Domingo). Adicionalmente, este proyecto va dirigido a los pacientes institucionalizados en residencias para personas mayores privadas y públicas de toda España (Centros Francisco de Vitoria y Cisneros de la Consejería de Políticas Sociales y Familia de la Comunidad de Madrid en España). También se incluirán personas residentes en la República Dominicana en áreas de con baja disponibilidad de recursos sanitarios (Carls George y Nuestra Señora del Carmen).

3.2.2 CitizenLab

El Proyecto CitizenLab (Entorno de experimentación y desarrollo de un modelo predictivo integral de comportamiento ciudadano individual y organizacional) busca proponer un entorno de experimentación y elaborar un modelo predictivo integral de comportamiento del ciudadano individual y colectivo que potencie además las zonas de la periferia. Teniendo en cuenta lo anterior, se plantea un conjunto estandarizado de conceptos, metodologías y criterios que permitan evaluar la situación actual y predecir comportamientos y estados futuros de los individuos de una población, de manera agregada y aislada, en relación con su estado de salud, hábitos de consumo, demandas de movilidad y necesidades de infraestructura y medio físico y que permitan un potenciamiento de las localidades objeto del proyecto orientado a los sectores de salud, automoción, turismo e infraestructuras.

4 Conclusiones y trabajos futuros

En esta tesis doctoral se ha demostrado diversos resultados obtenidos de los tres artículos principales y a partir de estos datos, podemos concluir que se cumplió satisfactoriamente con la hipótesis de investigación "el diseño arquitectónico enfocado a los microservicios mejora la eficiencia y el impacto de los servicios de eHealth, basándose en los criterios de aceptación", ya que se diseñó, desarrolló, implementó, desplegó y validó un diseño arquitectónico enfocado a los microservicios, aplicados en arquitecturas de software basadas en capas dentro del contexto eHealth, capaz de mejorar los servicios de salud desde el punto de vista de la gestión, la tecnología, la seguridad y la legalidad.

A continuación, se extrajo una serie de conclusiones más destacables a partir de los resultados obtenidos de los artículos publicados: (i) el primer artículo, se combinan dos dominios diferentes (salud y tecnología), no es fácil, pero los resultados son alentadores. Los clasificadores informaron de buenos resultados; el sistema ha sido bien aceptado por el personal médico y está demostrando ser rentable y una buena solución para atender a las zonas desfavorecidas o rurales. Teniendo en cuenta lo anterior, se explora la posibilidad de aplicar las comunicaciones móviles, los servicios en la nube y la tecnología de aprendizaje automático, con el fin de proporcionar herramientas eficientes para el personal médico en los hogares de ancianos; (ii) el segundo artículo, se emplea dos paradigmas computacionales (Edge & Cloud) para la implementación de una arquitectura de nube híbrida, con el fin de soportar aplicaciones versátiles de alto rendimiento, bajo el patrón de microservicios, para el prediagnóstico de enfermedades infecciosas en pacientes de la tercera edad. Los resultados obtenidos del análisis de la usabilidad del equipo médico en conjunto con el rendimiento de la arquitectura y el concepto de servicio muestran que la plataforma es factible e innovadora bajo la filosofía de alta cohesión y bajo acoplamiento; (iii) el tercer artículo, propone un nuevo marco de trabajo para la concepción de una plataforma eHealth centrada en los entornos de computación en la nube, basándose en los enfoques actuales (SOA y MSA) con respecto al desarrollo de sistemas de recomendación orientado a la telemonitorización y al acceso del historial clínico. Es por ello, que se realizaron las pruebas correspondientes, las cuales arrojaron que la variante MSA demuestra un mejor funcionamiento en cuanto al rendimiento y al tiempo de respuesta comparado a la variante SOA; sin embargo, el consumo de ancho de banda en MSA fue más significativo que SOA, y la escalabilidad en SOA generalmente no es posible o requiere mucho trabajo para lograrlo. En síntesis, consideramos que la implementación de SOA y MSA depende de la naturaleza y las necesidades de las organizaciones (i.e., rendimiento, disponibilidad o interoperabilidad) y para nuestra propias características de software MSA es la óptima en este aspecto bajo un escenario de salud.

En perspectiva, esta tesis aporta un fuerte argumento sobre la validez de los microservicios dentro del contexto eHealth, específicamente al telemonitoreo. Nos basamos en el concepto de aplicaciones versátiles de alto rendimiento y en los valores los valores cuantitativos que se obtuvieron de las diversas pruebas realizadas. Adicionalmente, hemos desarrollado un flujo de automatización para la integración y despliegue continuo de los microservicios. Dicho flujo, es adaptable a cualquier escenario médico (enfermedades infecciosas, cuidados paliativos, etc) y puede ser utilizado por un pequeño grupo de seis a ochos personas para el proceso de desarrollo de los servicios, con el objetivo de reducir el tiempo entre el compromiso de un cambio del sistema y colocar el cambio en producción.

Por otro lado, esta tesis da cabida a futuras investigaciones como combinar las mejores características SOA y MSA dentro de un ecosistema intergeneracional SOA-MSA, que conforme una suite de diferentes servicios, en lugar de un enfoque competitivo entre ellos [56], ya que permitirán que diferentes organizaciones médicas sin importar sus tecnologías tuvieran la capacidad de integrarse e interconectarse con diferentes aplicaciones eHealth adaptadas a las nuevas tecnologías, por ejemplo, sistema de inferencia basado en aprendizaje automático enfocada en realizar tareas especializadas en la prevención, monitoreo y tratamiento de las diversas afecciones. También, estamos explorando la posibilidad de extender nuestro marco de trabajo a otras áreas eHealth [54]–[56], como, por ejemplo: la predicción temprana de la hipotensión intradialítica durante sesiones de hemodiálisis, el pronóstico de la diabetes, la predicción de trastornos mentales en redes sociales y la predicción temprana de cáncer de colon mediante futuros proyectos de investigación y colaboración con el consorcio internacional que me encuentro afiliado durante mi estancia como estudiante de doctorado en la UAH.

Otra posible área de investigación futura sería la adaptación de esta propuesta hacia otros sectores de la industria 4.0 (p. ej. turismo, infraestructura, manufactura) [56]; sin embargo, se necesitan investigaciones adicionales para obtener suficientes resultados que demuestran la robustez del marco de trabajo, en cuanto a las adaptaciones de las características para dichos sectores, por lo cual tenemos una colaboración con el proyecto CitizenLab para cumplir con esta tarea futura y posteriormente se publicarán nuestros hallazgos en un futuro cercano.

5 Referencias

[1] M. Almiani, A. Abughazleh, Y. Jararweh, and A. Razaque, "Resilient Back Propagation Neural Network Security Model For Containerized Cloud Computing," *Simul. Model. Pract. Theory*, vol. 118, p. 102544, 2022, doi: https://doi.org/10.1016/j.simpat.2022.102544.

[2] V. Bushong *et al.*, "On Microservice Analysis and Architecture Evolution: A Systematic Mapping Study," *Appl. Sci.*, vol. 151, no. 17, p. 7856, Aug. 2019, doi: 10.3390/app11177856.

[3] I. Karabey Aksakalli, T. Çelik, A. B. Can, and B. Tekinerdoğan, "Deployment and communication patterns in microservice architectures: A systematic literature review," *J. Syst. Softw.*, vol. 180, p. 111014, 2021, doi: https://doi.org/10.1016/j.jss.2021.111014.

[4] Y. Yu, J. Liu, and J. Fang, "Online Microservice Orchestration for IoT via Multiobjective Deep Reinforcement Learning," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17513–17525, 2022, doi: 10.1109/JIOT.2022.3155598.

[5] F. Liu, J. Li, Y. Wang, and L. Li, "Kubestorage: A Cloud Native Storage Engine for Massive Small Files," in *2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC)*, 2019, pp. 1–4, doi: 10.1109/BESC48373.2019.8962995.

[6] J. Kosinska and K. J. Zielinski, "Experimental Evaluation of rule-based Autonomic Computing Management Framework for Cloud-native Applications," *IEEE Trans. Serv. Comput.*, p. 1, 2022, doi: 10.1109/TSC.2022.3159001.

[7] F. Tapia, M. Á. Mora, W. Fuertes, H. Aules, E. Flores, and T. Toulkeridis, "From Monolithic Systems to Microservices: A Comparative Study of Performance," *Applied Sciences* , vol. 10, no. 17. 2020, doi: 10.3390/app10175797.

[8] S. K. Lo, Q. Lu, L. Zhu, H.-Y. Paik, X. Xu, and C. Wang, "Architectural patterns for the design of federated learning systems," *J. Syst. Softw.*, vol. 191, p. 111357, 2022, doi: https://doi.org/10.1016/j.jss.2022.111357.

[9] F. Al-Doghman, N. Moustafa, I. Khalil, Z. Tari, and A. Zomaya, "AI-enabled Secure Microservices in Edge Computing: Opportunities and Challenges," *IEEE Trans. Serv. Comput.*, p. 1, 2022, doi: 10.1109/TSC.2022.3155447.

[10] S. Giallorenzo, F. Montesi, L. Safina, and S. P. Zingaro, "Ephemeral data handling in microservices with Tquery.," *PeerJ. Comput. Sci.*, vol. 8, p. e1037, 2022, doi: 10.7717/peerj-cs.1037.

[11] J. Spillner, P. Gkikopoulos, A. Buzachis, and M. Villari, "Rule-Based Resource Matchmaking for Composite Application Deployments across IoT-Fog-Cloud Continuums," in *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, 2020, pp. 336–341, doi: 10.1109/UCC48980.2020.00053.

[12] M. Sebrechts, B. Volckaert, F. De Turck, K. Yang, and M. Al-Naday, "Fog Native Architecture: Intent-Based Workflows to Take Cloud Native toward the Edge," *IEEE Commun. Mag.*, vol. 60, no. 8, pp. 44–50, 2022, doi:

10.1109/MCOM.003.2101075.
[13] R. Rezazadegan and M. Sharifzadeh, "Applications of Artificial Intelligence and Big Data in Industry 4.0 Technologies," in *Industry 4.0 Vision for Energy and Materials*, 2022, pp. 121–158.
[14] S. K. Jagatheesaperumal, M. Rahouti, K. Ahmad, A. Al-Fuqaha, and M. Guizani, "The Duo of Artificial Intelligence and Big Data for Industry 4.0: Applications, Techniques, Challenges, and Future Research Directions," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 12861–12885, 2022, doi: 10.1109/JIOT.2021.3139827.
[15] J. Sadek, D. Craig, and M. Trenell, "Design and Implementation of Medical Searching System Based on Microservices and Serverless Architectures," *Procedia Comput. Sci.*, vol. 196, pp. 615–622, 2022, doi: https://doi.org/10.1016/j.procs.2021.12.056.
[16] A. I. Awad, M. M. Fouda, M. M. Khashaba, E. R. Mohamed, and K. M. Hosny, "Utilization of mobile edge computing on the Internet of Medical Things: A survey," *ICT Express*, 2022, doi: https://doi.org/10.1016/j.icte.2022.05.006.
[17] M. A. P. da Silva, V. C. Times, A. M. C. de Araújo, and P. C. da Silva, "A Microservice-Based Approach for Increasing Software Reusability in Health Applications," in *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, 2019, pp. 1–8, doi: 10.1109/AICCSA47632.2019.9035229.
[18] K. Osei-Tutu and Y.-T. Song, "A Microservices Enterprise Architecture for Healthcare Information Exchange (HIE) in Developing Countries," in *2021 10th International Congress on Advanced Applied Informatics (IIAI-AAI)*, 2021, pp. 762–767, doi: 10.1109/IIAI-AAI53430.2021.00134.
[19] A. Mavrogiorgou *et al.*, "beHEALTHIER: A Microservices Platform for Analyzing and Exploiting Healthcare Data," in *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)*, 2021, pp. 283–288, doi: 10.1109/CBMS52027.2021.00078.
[20] M. Ianculescu, A. Alexandru, G. Neagu, and F. Pop, "Microservice-Based Approach to Enforce an IoHT Oriented Architecture," in *2019 E-Health and Bioengineering Conference (EHB)*, 2019, pp. 1–4, doi: 10.1109/EHB47216.2019.8970059.
[21] E. Enler, I. Pentek, and A. Adamko, "Healthcare Framework for Smarter Cities with bio-sensory data," in *2020 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, 2020, pp. 337–342, doi: 10.1109/CogInfoCom50765.2020.9237838.
[22] L. Rychener, F. Montet, and J. Hennebert, "Architecture Proposal for Machine Learning Based Industrial Process Monitoring," *Procedia Comput. Sci.*, vol. 170, pp. 648–655, 2020, doi: https://doi.org/10.1016/j.procs.2020.03.137.
[23] M. A. Jarwar, S. Ali, and I. Chong, "Exploring Web Objects enabled Data-Driven Microservices for E-Health Service Provision in IoT Environment," 2018, pp. 112–117, doi: 10.1109/ICTC.2018.8539684.
[24] C. Marques, V. Ramos, H. Peixoto, and J. Machado, "Pervasive Monitoring System for Services and Servers in Healthcare Environment," *Procedia Comput. Sci.*, vol. 201, pp. 720–725, 2022, doi:

https://doi.org/10.1016/j.procs.2022.03.097.
[25] E. Schäffer, H. Leibinger, A. Stamm, M. Brossog, and J. Franke, "Configuration based process and knowledge management by structuring the software landscape of global operating industrial enterprises with Microservices," *Procedia Manuf.*, vol. 24, pp. 86–93, Jan. 2018, doi: 10.1016/J.PROMFG.2018.06.013.
[26] T. Kiss *et al.*, "MiCADO—Microservice-based Cloud Application-level Dynamic Orchestrator," *Futur. Gener. Comput. Syst.*, vol. 94, pp. 937–946, May 2019, doi: 10.1016/J.FUTURE.2017.09.050.
[27] P. Di Francesco, P. Lago, and I. Malavolta, "Architecting with microservices: A systematic mapping study," *J. Syst. Softw.*, vol. 150, pp. 77–97, Apr. 2019, doi: 10.1016/J.JSS.2019.01.001.
[28] Y. Simmhan, P. Ravindra, S. Chaturvedi, M. Hegde, and R. Ballamajalu, "Towards a data-driven IoT software architecture for smart city utilities," *Softw. Pract. Exp.*, vol. 48, no. 7, pp. 1390–1416, Jul. 2018, doi: 10.1002/spe.2580.
[29] B. Götz, D. Schel, D. Bauer, C. Henkel, P. Einberger, and T. Bauernhansl, "Challenges of Production Microservices," *Procedia CIRP*, vol. 67, pp. 167–172, 2018, doi: 10.1016/J.PROCIR.2017.12.194.
[30] M. Krämer, S. Frese, and A. Kuijper, "Implementing secure applications in smart city clouds using microservices," *Futur. Gener. Comput. Syst.*, vol. 99, pp. 308–320, Oct. 2019, doi: 10.1016/J.FUTURE.2019.04.042.
[31] C. Gutiérrez and M. López, "La salud en la era digital," *Rev. Médica Clínica Las Condes*, vol. 33, no. 6, pp. 562–567, 2022, doi: https://doi.org/10.1016/j.rmclc.2022.11.001.
[32] J. M. Rosa and E. L. Frutos, "Ciencia de datos en salud: desafíos y oportunidades en América Latina," *Rev. Médica Clínica Las Condes*, vol. 33, no. 6, pp. 591–597, 2022, doi: https://doi.org/10.1016/j.rmclc.2022.09.007.
[33] Y. Liu, W. Ren, Y. Qiu, J. Liu, P. Yin, and J. Ren, "The Use of Mobile Phone and Medical Apps among General Practitioners in Hangzhou City, Eastern China," *JMIR mHealth uHealth*, vol. 4, no. 2, p. e64, 2016, doi: 10.2196/mhealth.4508.
[34] L. Fan *et al.*, "DACAR Platform for eHealth Services Cloud," in *2011 IEEE 4th International Conference on Cloud Computing*, 2011, pp. 219–226, doi: 10.1109/CLOUD.2011.31.
[35] P. Souza, B. Hochhegger, J. Cavalcanti da Silveira, C. Pretto, C. Moralles, and A. Tronchoni, *MediCloud – Telemedicine System Based on Software as a Service (SaaS)*. 2015.
[36] J. Keen, "Digital health care: Cementing centralisation?," *Health Informatics J.*, vol. 20, no. 3, pp. 168–175, Sep. 2014, doi: 10.1177/1460458213494033.
[37] F. Kargl, E. Lawrence, M. Fischer, and Y. Y. Lim, "Security, Privacy and Legal Issues in Pervasive eHealth Monitoring Systems," in *2008 7th International Conference on Mobile Business*, 2008, pp. 296–304, doi: 10.1109/ICMB.2008.31.
[38] H. Calderón-Gómez *et al.*, "Proposal Using the Cloud Architecture in System for the Early Detection of Infectious Diseases in Elderly People

Fed by Biosensors Records," in *2019 7th International Engineering, Sciences and Technology Conference (IESTEC)*, 2019, pp. 631–634, doi: 10.1109/IESTEC46403.2019.00118.
[39] H. Calderón-Gómez *et al.*, "Development of ehealth applications-based on microservices in a cloud architecture," *RISTI - Rev. Iber. Sist. e Tecnol. Inf.*, vol. 2019, no. E23, pp. 81–93, Oct. 2019.
[40] J. Sanz-Moreno *et al.*, "mHealth System for the Early Detection of Infectious Diseases Using Biomedical Signals," in *Proceedings of the Latin American Congress on Automation and Robotics*, 2020, pp. 203–213, [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-030-40309-6_20.
[41] L. Chen, "Microservices: Architecting for Continuous Delivery and DevOps," in *2018 IEEE International Conference on Software Architecture (ICSA)*, 2018, pp. 39–397, doi: 10.1109/ICSA.2018.00013.
[42] V. Kontis, J. E. Bennett, C. D. Mathers, G. Li, K. Foreman, and M. Ezzati, "Future life expectancy in 35 industrialised countries: projections with a Bayesian model ensemble.," *Lancet (London, England)*, vol. 389, no. 10076, pp. 1323–1335, Apr. 2017, doi: 10.1016/S0140-6736(16)32381-9.
[43] M. Solé-Casals, E. Chirveches-Pérez, E. Puigoriol-Juvanteny, N. Nubó-Puntí, C. Chabrera-Sanz, and M. Subirana-Casacuberta, "Perfil y resultados del paciente frágil valorado por la Enfermera de Práctica Avanzada en un servicio de urgencias," *Enfermería Clínica*, vol. 28, no. 6, pp. 365–374, 2018, doi: https://doi.org/10.1016/j.enfcli.2017.04.003.
[44] A. Baldominos, H. Oğul, and R. Colomo-Palacios, "Infection Diagnosis using Biomedical Signals in Small Data Scenarios," in *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, 2019, pp. 38–43, doi: 10.1109/CBMS.2019.00018.
[45] M. E. García Leoni, B. Macías Bou, L. Martín González, and E. Martínez Larrull, "Infecciones respiratorias," *Medicine (Baltimore).*, vol. 10, no. 88, pp. 5947–5954, Oct. 2011, doi: 10.1016/S0304-5412(11)70203-7.
[46] S. G. Remartínez, M. G. Pión, F. J. G. Gómez, and E. G. García, "Infecciones respiratorias en Urgencias," *Med. - Programa Form. Médica Contin. Acreditado*, vol. 11, no. 88, pp. 5254–5263, 2015, doi: https://doi.org/10.1016/j.med.2015.10.007.
[47] D. Graziani Noriega and A. Ampuero López, "Protocolo diagnóstico y tratamiento empírico en urgencias de las infecciones respiratorias," *Med. - Programa Form. Médica Contin. Acreditado*, vol. 12, no. 64, pp. 3794–3800, 2018, doi: https://doi.org/10.1016/j.med.2018.10.008.
[48] J. R. Yuste Ara, J. L. del Pozo, and F. Carmona-Torre, "Protocolo de tratamiento empírico de las infecciones urinarias," *Med. - Programa Form. Médica Contin. Acreditado*, vol. 12, no. 51, pp. 3041–3047, 2018, doi: https://doi.org/10.1016/j.med.2018.03.007.
[49] J. Medina-Polo *et al.*, "Infecciones urinarias adquiridas en la comunidad que requieren hospitalización: factores de riesgo, características microbiológicas y resistencia a antibióticos," *Actas Urológicas Españolas*, vol. 39, no. 2, pp. 104–111, 2015, doi: https://doi.org/10.1016/j.acuro.2014.08.001.
[50] R. Khan and M. Amjad, "Performance testing (load) of web applications

based on test case management," *Perspect. Sci.*, vol. 8, pp. 355–357, Sep. 2016, doi: 10.1016/J.PISC.2016.04.073.
[51] R. Ramakrishnan and A. Kaur, "Little's law based validation framework for load testing," *Inf. Softw. Technol.*, vol. 108, pp. 88–98, Apr. 2019, doi: 10.1016/J.INFSOF.2018.11.007.
[52] Universidad de Alcalá, "Guía para la elaboración y la defensa de la Tesis Doctoral." Alcala de Henares, p. 15, 2022, [Online]. Available: https://escuela-doctorado.uah.es//tesis/documentos/guia_tesis.pdf.
[53] Universidad de Alcalá, "REGLAMENTO DE ELABORACIÓN, AUTORIZACIÓN Y DEFENSA DE LA TESIS DOCTORAL." Alcala de Henares, pp. 1–9, 2016, [Online]. Available: https://escuela-doctorado.uah.es/escuela/documentos/reglamento_tesis.pdf.
[54] A. Garcés-Jiménez *et al.*, "Medical Prognosis of Infectious Diseases in Nursing Homes by Applying Machine Learning on Clinical Data Collected in Cloud Microservices," *International Journal of Environmental Research and Public Health* , vol. 18, no. 24. 2021, doi: 10.3390/ijerph182413278.
[55] H. Calderón-Gómez *et al.*, "Telemonitoring System for Infectious Disease Prediction in Elderly People Based on a Novel Microservice Architecture," *IEEE Access*, vol. 8, pp. 118340–118354, 2020, doi: 10.1109/ACCESS.2020.3005638.
[56] H. Calderón-Gómez *et al.*, "Evaluating Service-Oriented and Microservice Architecture Patterns to Deploy eHealth Applications in Cloud Computing Environment," *Applied Sciences* , vol. 11, no. 10. 2021, doi: 10.3390/app11104350.