



Propuesta de arquitectura de microservicios, metodología Scrum para una aplicación móvil de control académico: Caso Escuela Profesional de Obstetricia de la Universidad Nacional Mayor de San Marcos

Microservices architecture proposal, Scrum methodology for a mobile application of academic control: Case of the Universidad Nacional Mayor de San Marcos Professional School of Obstetrics

Percy Edwin De la Cruz Vélez de Villa¹
<https://orcid.org/0000-0002-4943-7620>

Maycol Henry Espinoza Ramírez²
Omar Cuba Estrella³

Universidad Nacional Mayor de San Marcos, Perú

Recibido: 14-04-2019

Aceptado: 12-08-2019

CITA RECOMENDADA

De la Cruz, P., Espinoza, M. & Cuba, O. (2019). Propuesta de arquitectura de microservicios, metodología Scrum para una aplicación móvil de control académico: Caso Escuela Profesional de Obstetricia de la Universidad Nacional Mayor de San Marcos. Hamut'ay, 6(2), 141-158.
<http://dx.doi.org/10.21503/hamu.v6i2.1781>

RESUMEN

El objetivo de este documento es presentar una propuesta de un sistema móvil para mejorar la gestión y control académico en la Escuela Profesional de Obstetricia de la Universidad Nacional Mayor de San Marcos. Las fases planteadas del estudio fueron tres. La primera fue de planificación y análisis, relacionadas con la metodología Scrum, donde se visualiza cómo se gestiona el proyecto con la participación de los usuarios. La segunda de diseño, donde se desarrolló el marco de la arquitectura de microservicios apreciándose su aplicación al buscar unidades funcionales mínimas para su operatividad e independencia. Esta integración permitió la realización de la propuesta para el adecuado seguimiento y control a docentes y estudiantes así cómo, para apoyar la gestión académica. El uso del móvil como instrumento de entrada y salida en este proyecto da un cierto nivel de ubicuidad, permitiendo según rol (alumno, docente o

¹ Docente Principal, director de la Escuela Profesional de Ingeniería de Sistemas. Licenciado en Computación, Ingeniero de Sistemas, Mg. En Computación e Informática, Estudios doctorales concluidos en Ciencias Administrativas, Ingeniería de Sistemas e Informática. Consultor Informático en TI. Experiencia profesional en empresas del sector Industrial, Pesquero, Metal mecánica. Línea de investigación: Gestión del conocimiento, Big data-Minería de datos, Ingeniería de software. E-mail: pdelacruz@unmsm.edu.pe

² Bachiller en Ingeniería de sistemas, FISI-UNMSM. Ejerce como Ingeniero de Software. Experiencia profesional en proyectos estatales y privados, en la UNMSM (Quipucamayoc), Banco Falabella, Entel, Profuturo AFP, Interbank. Conocimiento de programación Java, servicios web, arquitecturas monolíticas, microservicios, Scrum, RUP, Oracle Suite, Mysql, Spring, Java, GIT, SVN, Jenkins, Docker, AWS Cloud, React Native, Javascript. E-mail: mhersch1@gmail.com

³ Bachiller en Ingeniería de sistemas, FISI-UNMSM. Se desempeña como analista programador en java. Experiencia profesional en proyectos estatales en la UNMSM (Quipucamayoc), Centro de Producción FISI, RENIEC. Conocimiento de: Programación Java, desarrollo web, arquitecturas monolíticas, microservicios, Scrum, RUP, ORACLE Suite, Mysql, Spring, GIT, SVN. E-mail: cuba.omar13@gmail.com



director) usar las funcionalidades y lograr esa integración y explotación de la información orientadas al logro de los estándares 14, 20 y 30 exigidos por el Sistema Nacional de Evaluación, Acreditación y Certificación de la Calidad Educativa de Perú.

Palabras clave: calidad educativa, control académico, microservicios, Scrum.

ABSTRACT

The objective of this document is to present a proposal for a mobile system to improve the management and academic control at the Professional School of Obstetrics of the San Marcos National University. The proposed phases of the study were three. The first one was related to planning and analysis, related to the Scrum methodology, which shows how the project is managed with the participation of the users. The second phase is related to design, where the microservices architecture framework was developed, observing its application in terms of looking for minimum functional units for its operability and independence. This integration allowed the implementation of the proposal for the adequate monitoring and control of teachers and students as well as, to support the academic management. The use of the mobile as an input and output instrument in this project gives a certain level of ubiquity, allowing according to the role (student, teacher or manager) to use the functionalities and achieve that integration and handling of the information oriented to the achievement of standards 14, 20 and 30 required by the National System of Assessment, Accreditation and Certification of Educational Quality of Peru.

Keywords: educational quality, academic control, microservices, Scrum.

INTRODUCCIÓN

Según rankings internacionales de las 100 primeras universidades, ninguna universidad peruana se encuentra ubicada entre las 30 primeras (Quacquarelli, 2011). Otros indicadores estadísticos como COMEXPERU (2011), manifiestan una desigualdad de oportunidades para el acceso a la educación universitaria, producto del centralismo con respecto a las provincias manifestándose una desigualdad de acceso y oportunidades, asimismo, la creación de muchas y diversas universidades privadas que no cumplen con los estándares establecidos para el proceso de licenciamiento de la Superintendencia Nacional de Educación Superior Universitaria (SUNEDU), por lo cual para garantizar un buen nivel competitivo, se exige que primero sean licenciadas por SUNEDU y posteriormente optar por una acreditación. En concor-

dancia con la publicación de SINEACE (Yanada, Rivera, & Castro, 2012), el Perú se encontraba por debajo del rango promedio con lo que respecta a calidad educativa en comparación con otros países, siendo evidente que había problemas en la educación universitaria en Perú. Quacquarelli (2019) manifiesta que del ranking de las 100 mejores universidades latinoamericanas las mejores ubicadas son, en el puesto 21 Pontificia Universidad Católica del Perú, en el puesto 70 Universidad Peruana Cayetano Heredia, y en el puesto 74 Universidad Nacional Mayor de San Marcos. Con la finalidad de revertir estas dificultades es creado el Sistema Nacional de Evaluación, Acreditación y Certificación de la Calidad Educativa (SINEACE) para apoyar la mejora de la calidad universitaria mediante estándares. La Facultad de Medicina de la Universidad Nacional Mayor de San Marcos (UNMSM) desde la Escuela Profesio-

nal de Obstetricia (EPO) es consciente de que se requiere cumplir con los estándares para continuar posicionándose dentro del ámbito académico, y al tener ciertas falencias que han mermado en una calidad no idónea en algunas de sus áreas, como la información de las clases referidas al registro de notas, la toma de asistencia, la cual se registra en diferentes medios como: papel físico, archivo Excel u otros, no se cuenta con un sistema centralizado, que permita disponer con dicha información, generando retrasos en conseguir y entregar la información entre el profesor y los alumnos. Para superar estos inconvenientes algunos profesores acuden a aplicaciones informáticas (correo, Facebook, WhatsApp, etc.) para comunicar avisos, tales como, el registro de notas, la lista de no asistencia, falta a clase por enfermedad, comunicados de postergación de clase por feriado u otros acontecimientos, lo cual causa pérdida de tiempo, genera insatisfacción al no tener todos los estudiantes el acceso de esta información para ver sus avances académicos, y en muchos casos que ellos no puedan utilizar en sus hogares esta tecnología, pero, lo más álgido por lo general, es que no se llega a cumplir todos los temarios del sílabo de los cursos en su totalidad. No se puede hacer un seguimiento a todas estas problemáticas, ya que la información impartida en las clases, la ejecución curricular (sílabo), las asistencias de alumnos y profesores, y las notas se formalizan y se registran a fin de ciclo y no está a disposición para su uso y transferencia.

Los aspectos descritos líneas arriba motivaron a que se realice la propuesta del diseño de un sistema distribuido móvil que abarque el control académico y la gestión en la EPO. Se usará la metodología Scrum para la planificación y análisis del proyecto que exige la participación activa de los usuarios en su proceso de desarrollo (Fernández & Cadelli, 2014), por otro lado, para el diseño nos apoyaremos con la arquitectura de microservicios, que tiene estructura modular, independiente, fáciles de implementar, permitiendo alta cohesión y bajo acoplamiento.

MÉTODO

Para el fundamento de la propuesta se hizo una

revisión de la literatura, para lo cual se seleccionó material bibliográfico y documental de libros digitales relacionado a arquitecturas SOA y microservicios, así como cada uno de los componentes que la conforman, en repositorios como: Scielo, ACM, IEEE, Redalyc, Scopus y WOS.

Para la búsqueda se utilizaron descriptores como: Arquitecturas de software: SOA y microservicios, sistema de Control, Performance Management Control (PMC), Applied Computing, Enterprise Computing, Service-oriented Architecture, Computers and information processing, Distributed computing y distributed information system. Para la selección de búsqueda se tuvo como criterio de inclusión, que los artículos sean de los últimos 5 años.

Sistema Nacional de Evaluación, Acreditación y Certificación de la Calidad

En el año 2006 se decreta la ley 28740 referida a la creación del Sistema Nacional de Evaluación, Acreditación y Certificación de la Calidad (SINEACE) cuyo objetivo es apoyar a las instituciones educativas a conseguir la acreditación mediante un proceso de evaluación (SINEACE, 2016). En el año 2014, El modelo de acreditación de SINEACE es reorganizado debido a la creación de la Ley universitaria 30220. Este nuevo modelo se basa en una matriz de 34 estándares, de los cuales se han tomado en cuenta tres de ellos, tabla 1, debido a que solo estos están relacionados y pertenecen a la parte académica que corresponde a este estudio.

Tabla 1

Tres estándares del nuevo modelo SINEACE, según Ley universitaria 30220

Estándar	Descripción
Estándar 14. Selección, evaluación, capacitación y perfeccionamiento	El programa de estudios debe evaluar el desempeño del personal docente con el fin de identificar necesidades para capacitaciones. El perfeccionamiento se refiere a actualización, innovación pedagógica y manejo de tecnologías de información de los profesores.

Estándar	Descripción
Estándar 20. Seguimiento al desempeño de los estudiantes.	El programa de estudios debe contemplar el seguimiento del desempeño de los estudiantes y ofrecer apoyo como servicios de tutoría o reforzamiento en cursos.
Estándar 30. Sistema de información y comunicación	El programa de estudios debe tener implementado un sistema de información y comunicación que apoye a la gestión académica y administrativa.

Fuente: Adaptado de SINEACE (2016).

Según Moromi (2002), la calidad académica universitaria esta engranada con una serie de factores que se tornan necesarios para que esta sea reconocida a nivel académico, profesional y empresarial, siendo uno de ellos la ejecución curricular (control o seguimiento del sílabo o currículo del curso) en el rendimiento académico de los estudiantes universitarios. Asimismo, demuestra la importancia que tiene el cumplimiento del sílabo en la aprobación de una asignatura, y como esta se relaciona directamente entre la ejecución curricular y el rendimiento académico, y, además, resalta la correlación entre la percepción positiva que tienen los alumnos con la ejecución curricular con un mejor rendimiento académico. Otro factor viene a ser la asistencia a clases, según Rodríguez & Herrera, (2009), la asistencia a clases es un factor determinante para obtener el rendimiento académico esperado. Como conclusión del estudio realizado por estos autores, se demuestra que la asistencia a clases tanto práctica como teórica influye en la superación o mejor rendimiento académico en los estudiantes universitarios. Otro de los aspectos importantes es el desempeño académico del docente con base en los alumnos; según Sifuentes (2018), en su estudio indica que hay una relación significativa en el desempeño académico del docente universitario con la satisfacción de los estudiantes, incide en el rendimiento académico. Para Tolentino (2014) menciona que hay una relación significativa entre el desempeño docente universitario y la satisfacción de los estudiantes del programa.

Sistemas de control de gestión

Horngren, Datar & Foster, (2000), definen al sis-

tema de control de gestión como “Un sistema que adquiere y usa información que ayuda a la coordinación del planeamiento y las decisiones del control organizacional con el objetivo de mejorar las decisiones colectivas dentro de la organización”. Por su parte Abernethy & Chua, (1996), indican que es un sistema que combina mecanismos de control diseñados e implementados por los administradores para incrementar la probabilidad de que los líderes de la organización se comporten de una manera consistente con los objetivos de esta. Por lo tanto, un Sistema de control de Gestión debe ser capaz de medir y gestionar el desempeño organizacional, alineados con la generación de valor para la organización, particularmente dirigido a grupos que desempeñen un papel estratégico. La necesidad de controlar el desempeño basado en los flujos de información consistente es la motivación de ser en los sistemas de control de gestión.

La evaluación del desempeño organizacional es trascendental para cualquier tipo de organización, uno de los motivos más importantes por el que las organizaciones deben implementar un sistema de evaluación y control de gestión de sus recursos humanos, es para saber si sus trabajadores están efectivamente contribuyendo al logro de los objetivos institucionales establecidos (Sánchez, 2011), al ser un conjunto de medidas que tratan de estar alineadas con la estrategia de la organización. Además, busca la satisfacción de los stakeholders e influye en todos los actores del negocio a mejorar sus actividades integrando la estrategia, los procesos y los recursos.

Una de las funciones de la evaluación de desempeño es la de proveer información valiosa para la toma de decisiones, siendo este el apoyo principal para el proceso de planeamiento y control, así como mantenerse alineados con los objetivos trazados de la organización. Otra importante funcionalidad es la “señalización”, esto quiere decir, mostrar a los empleados la importancia de los aspectos estratégicos establecidos y brindar información no financiera a los stakeholders, tales como la innovación, operaciones, niveles de satisfacción del cliente, entre otros.

Según Che & Rapih, (2013) el Sistema de Control de Gestión (CMS) facilita la gestión de los

entornos crecientes en complejidad. Por otro lado, los Sistemas de Gestión de Rendimiento (PMS) son más horizontales; orientado al proceso y centrado en las necesidades de los grupos de interés. Consecuentemente, es interesante considerar MCS en el diseño de PMS para superar la debilidad del PMS tradicional y mejorar en general el rendimiento.

Para Ferreira & Otley, (2006) una forma de evaluar a los sistemas de control de gestión es a través del framework “Gestión del desempeño y control”, cuya finalidad es capturar la estructura y funcionalidad de los sistemas de control de gestión. A continuación, Tabla 2, se listan los 12 tópicos centrales y las preguntas con las que éstas son formuladas:

Tabla 2
Tópicos centrales y preguntas de sistema de control de gestión

Tópicos	Preguntas
Misión y Visión	¿Cuál es la visión y misión de la organización y como se llama la atención de los gerentes y empleados? ¿Qué mecanismos, procesos y redes son usados para transmitir los propósitos y objetivos generales de la organización hacia sus miembros?
Factores claves de éxito	¿Cuáles son los factores claves que se consideran fundamentales para el éxito general de la organización y como les llama la atención a los gerentes y empleados?
Planes y estrategias	¿Cuál es la estructura de la organización y que impacto tiene en el diseño y uso de los sistemas de gestión del rendimiento? ¿Cómo influye y como está influenciado por el proceso de gestión estratégica?
Estructura de la organización	¿Cuáles son las medidas de desempeño clave de la organización?
Indicadores claves del desempeño	¿Cuáles son las medidas de desempeño clave de la organización que se derivan de sus objetivos, factores clave de éxito y estrategia y planes? ¿Cómo se especifican y qué papel juegan en la evaluación de desempeño? ¿Existen omisiones significativas?
Establecimiento de objetivos	¿Qué nivel de desempeño debe alcanzar la organización para cada una de sus medidas de desempeño clave (identificadas en la pregunta anterior),

Tópicos	Preguntas
	como se encarga de establecer los objetivos de desempeño apropiados para ellos y que tan difíciles son esos objetivos?
Evaluación del desempeño	¿Qué procesos, si los hay, sigue la organización para evaluar el desempeño individual, grupal y organizacional? ¿Las evaluaciones de desempeño son principalmente objetivas, subjetivas o mixtas y que tan importantes son la información y los controles formales e informales en estos procesos?
Sistema de recomendaciones	¿Qué procesos, si los hay, sigue la organización para evaluar el desempeño individual, grupal y organizacional? ¿Las evaluaciones de desempeño son principalmente objetivas, subjetivas o mixtas y que tan importantes son la información y los controles formales e informales en estos procesos?
Flujos de información	¿Qué flujos específicos de información (retroalimentación y avance), sistemas y redes tiene implementada la organización para respaldar el funcionamiento de sus sistemas de gestión de desempeño?
Tipos de uso de los flujos	¿Qué tipo de uso se hace de la información y de los diversos mecanismos de control establecidos? ¿Se pueden caracterizar estos usos en términos de diversas tipologías en la literatura? ¿Cómo difieren los controles y sus usos en diferentes niveles jerárquicos?
Cambios en el Sistema PMC	¿Cómo se han modificado los PMS a la luz de las dinámicas de cambio de la organización y su entorno? ¿Se han realizado los cambios en el diseño o uso de PMS de manera proactiva o reactiva?
Fuerza y consistencia de los componentes	¿Qué tan fuertes y coherentes son los vínculos entre los componentes de los PMS y las formas en que se utilizan (como indica en las 11 preguntas anteriores)?

Fuente: Adaptado de (Ferreira & Otley, 2006) citado en (Beuren & Teixeira, 2014).

Aunque existen otros tipos de metodologías como el Proceso Unificado Racional (RUP), éste es una metodología estándar más usada para el análisis, implementación y documentación de sistemas orientados a objetos, de acuerdo a Fernández &

Cadelli, 2014. Luego de realizar las comparaciones, en este estudio se eligió aplicar la metodología Scrum.

Scrum

Según Fernández & Cadelli, (2014), Scrum es una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo. Pone su atención y hace foco sobre valores y prácticas de gestión. Delega completamente al equipo la responsabilidad de decidir la mejor manera de trabajar para ser lo más productivos posibles, es decir, es flexible y los integrantes del equipo pueden optar por organizar la forma de interactuar entre ellos. Toma el cambio como una forma de entregar al final del desarrollo algo más cercano a la verdadera necesidad del cliente. Como metodología ágil está basado en los siguientes principios ágiles:

Tabla 3

Principios de manifiesto ágil

Principios de Manifiesto Ágil
Satisfacer al cliente mediante tempranas y continuas entregas de software que aporten valor. Entregar software funcional lo más pronto posible.
Predisposición y respuesta al cambio dado que presentan una ventaja competitiva.
Entregar frecuentemente software que funcione y no planificaciones o documentación de análisis o diseño.
La gente del negocio y los desarrolladores deben trabajar juntos, el proceso de negocio debe ser guiado por el cliente por lo que interactuar con el equipo es frecuente.
El proyecto debe contar con individuos motivados y brindarles apoyo.
Fortalecer la comunicación y la colaboración.
El software que funciona es la medida del progreso.
El desarrollo debe ser sostenible por los desarrolladores y usuarios deben colaborar para que esto suceda.
Atención continua en la calidad técnica y buen diseño mejora la agilidad.
La simplicidad es esencial, camino simple consistente con objetivos claros.
El equipo debe ser auto organizado, ellos determinan los objetivos que persiguen.
Reflexión sobre cómo llegar a ser más efectivo en intervalos de tiempo, adaptarse a nuevos entornos.

Fuente: Adaptado de Fernández & Cadelli, (2014) y Britto, (2016).

El flujo de trabajo de un Scrum, Figura 1, está formado por una serie de elementos que según Fernández & Cadelli, (2014), están compuestos por el Product Backlog o lista completa de requerimientos priorizados, que cambia constantemente, no llega a ser totalmente definida; cumple roles, de un Product Owner, que representa a todos los interesados del producto final; además gestiona los requerimientos del Product Backlog. Posee un Scrum Master, responsable del proceso Scrum, que asegura el desarrollo del proyecto de acuerdo con las prácticas, valores y reglas de Scrum y que progresa según lo previsto. Interactúa con el cliente y el equipo. Encargado de realizar reuniones. En cambio el equipo Scrum es responsable de transformar el Backlog de la iteración en un incremento de la funcionalidad del software (entregable de software).

El Sprint Backlog otro de los elementos hace referencia al trabajo o tareas determinadas por el equipo para realizar en un Sprint. Es decir, la conversión de tareas a un producto funcional. Teniendo como características, que las tareas se estiman en una duración entre 1 a 20 horas de trabajo. Las de mayor duración deben intentar descomponerse en sub-tareas de ese rango de tiempo y que la estimación sea actualizada día a día.

El Sprint, es el período de tiempo durante el que se desarrolla un incremento de funcionalidad; las características del Sprint son: su duración que es de 1 a 4 semanas. Durante el desarrollo del Sprint no se puede modificar el trabajo que se ha acordado en el Backlog, sólo es posible cambiar el curso de un Sprint, abortándolo, y sólo lo puede hacer el Scrum Master si decide que no es viable por alguna de estas razones; la tecnología acordada no funciona, o las circunstancias del negocio han cambiado, o el equipo ha tenido interferencias. El Scrum diario, es una reunión dada cada 24 horas por los integrantes del equipo para determinar el avance de tareas y detectar problemas que impidan no cumplir con las metas del sprint. Para cumplir con lo programado la revisión del Sprint; tiene como fin presentar el producto desarrollado a los usuarios, se analizan y se detectan inconformidades las cuales se pueden planificar y ver en el siguiente sprint. Como último elemento esta la reunión retrospectiva del Sprint, es la reunión del

equipo Scrum, Product Owner y Scrum Master para ver puntos positivos y negativos del sprint.

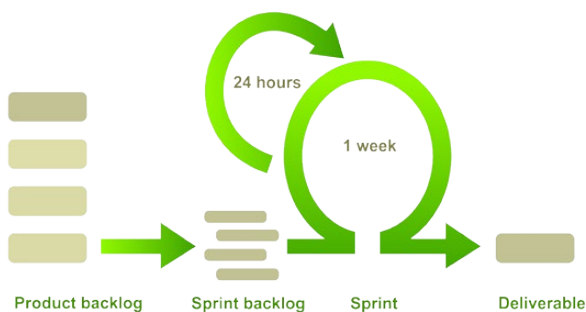


Figura 1.
Flujo de trabajo de SCRUM.
Fuente: Fernández & Cadelli, (2014)

Arquitectura Orientada a Servicios

Según Erl, (2016) la Arquitectura Orientada a Servicios (SOA) es un modelo de arquitectura que posiciona a los servicios como un mecanismo esencial para poder incrementar la eficiencia, agilidad y productividad de una empresa. La implementación de SOA puede consistir de la combinación de tecnologías, productos, APIs, extensiones para el soporte de una infraestructura, y otras partes.

Los servicios vienen a ser los componentes unitarios de la arquitectura SOA, son aplicaciones que residen en servidores centralizados, éstos intercambian datos entre sistemas independientemente del lenguaje de programación en el que estén desarrolladas y de la plataforma en dónde se ejecuten. Según Krafzig, Banke, & Slama, (2004, pp. 58-65), SOA tiene una serie de elementos como:

- i. Aplicación Frontend, es uno de los artefactos de SOA que inicia y controla toda la actividad de los sistemas empresariales.
- ii. Servicios, componente de software de significado funcional distinto que típicamente encapsula un concepto de negocio de alto nivel y contiene:
 - A. El contrato del servicio que provee una especificación informal del propósito, funcionalidad, restricciones, y uso del servicio. La forma de esta especificación puede

variar, dependiendo del tipo de servicio.

- B. La conexión basada en lenguajes como IDL o WSDL. Provee la abstracción e independencia de la tecnología, incluyendo el lenguaje de programación, middleware, protocolo de red y entorno de ejecución. Además, puede imponer detalles semánticos de la funcionalidad y parámetros que no están sujetos a IDL o especificaciones WSDL. La interfaz es expuesta por la conexión del servicio a los clientes que son conectados al servicio usando una red. La implementación de esta interacción denominada “stub service” se usa para desarrollo y pruebas.
- C. Implementación del servicio, físicamente provee la lógica del negocio requerido y los datos apropiados. Contiene a artefactos tales como programas, datos de configuración y bases de datos. Compuesto por:
 - c.1 Lógica de Negocio: es encapsulado por un servicio, es parte de su implementación. Se pone a disposición a través de las interfaces de servicio.
 - c.2 Data, un servicio que puede también incluir datos.

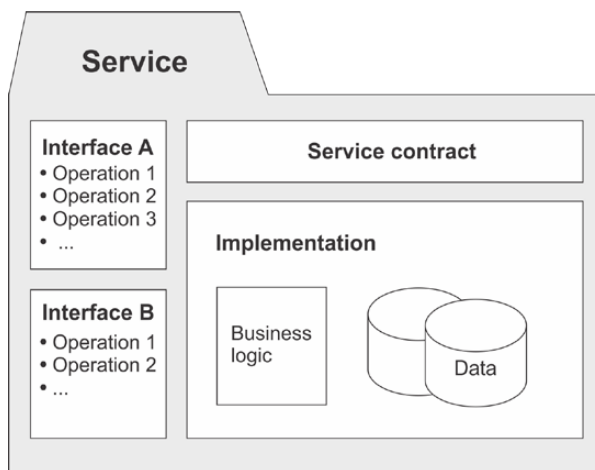


Figura 2.
Componentes de un servicio en SOA.
Fuente: (Krafzig, Banke, & Slama, 2004)

- iii. Servicio de Repositorio: Un repositorio de servicios provee facilidades para descubrir servicios y adquirir toda la información para usar los servicios.

- iv. Bus de Servicios: conecta a todos los servicios SOA y aplicaciones frontend.

Arquitectura de Microservicios

Según Perez-Herrera, (2015) refiere que este tipo de arquitectura tiene como idea principal dividir el software en servicios que sean los más independientes posibles entre ellos distribuidos en diferentes lugares lo cual lo convierte en un sistema distribuido. Según Fowler & Lewis, (s.f.), los microservicios se definen como un estilo arquitectural en el que múltiples servicios funcionan individualmente desplegados de una manera automatizada, además se comunican con mecanismos ligeros generalmente usando un recurso API basado en HTTP. Según Newman, (2015, pp. 2-3), los microservicios son servicios autónomos que trabajan juntos. Menciona que los aplicativos monolíticos usan un código fuente en muchos casos muy extenso, lo cual dificulta hacer nuevos cambios o el corregir errores sea más difícil, en contraparte a microservicios que persigue. Para esto necesita ser cohesivo, es decir, agrupar código con la misma razón y separar por otro lado código que cumpla diferentes razones. Es por eso que microservicios se enfocan en hacer servicios con base en los límites del negocio, deben ser lo suficientemente pequeños, porque mientras más se reduzca el tamaño más crecen las ventajas. Cada microservicio debe ser desplegado de manera aislada en una plataforma como servicio (PAAS), es por eso que se trata de evitar que múltiples servicios estén en un solo lugar. Las comunicaciones entre estos son mediante llamadas en red. Newman (2015, pp. 45-50), menciona que se puede escoger que principios adoptar en base a la organización, siempre conociendo los riesgos que ello aqueja, en la Tabla 4 se mencionan los principios más resaltantes.

Tabla 4
 Principios de arquitectura de microservicios

Principios	Descripción
Modelar entorno a conceptos de negocios	Que en base a la experiencia de interfaces basadas en límites del contexto de negocio son más estables. Al modelar en dominios se asegura que se reflejen los cambios de los procesos de negocio más fácilmente.

Principios	Descripción
Adoptar la cultura de automatización	Dado que los microservicios aumentan la complejidad, es bueno contar con fortalecer una cultura de automatización con herramientas que soporten los microservicios. El testeo automatizado es esencial, es un proceso más complejo que los sistemas monolíticos, también se menciona que se debe adoptar la integración continua para una rápida retroalimentación en la calidad de la producción.
Esconder detalles internos de implementación	Para maximizar la evolución de los microservicios independientes, es vital que se esconda los detalles de su implementación y definir bien los límites de negocios es un buen soporte. Los servicios deben esconder su base de datos para evitar el acoplamiento y para la generación de reportes hacer un movimiento masivo de datos por eventos. Los servicios deben proveer APIS con tecnología agnóstica (que son servicios que no se preocupan del contexto en el que son llamados) que permite que se usen diferentes tecnologías. Considerar usar REST que formaliza la separación de la implementación interna y externa.
Descentralizar todas las cosas	Para maximizar la autonomía de los microservicios se necesita buscar constantemente los cambios para delegar el control de los servicios a los equipos. Asegurar que cada equipo tiene sus servicios es un paso importante, son responsables de los cambios, de cuando lanzan una nueva versión, cada equipo debe ser experto en el dominio del negocio en el cual están creando. Intentar adoptar un modelo de gobierno compartido en los que cada equipo colectivamente comparta responsabilidades de la visión técnica del sistema. La arquitectura, puede evitar enfoques como los buses de servicios empresariales o sistemas de orquestación los cuales centralizan la lógica de negocios y sus servicios son tontos. En vez preferir coreografía sobre orquestación y usar un middleware tonto con endpoint inteligentes que aseguren mantener la lógica asociada y su información dentro de los límites de los servicios ayudando a tener las cosas cohesivas.
Despliegue independiente	Se debe tratar de asegurar que los microservicios puedan ser desplegados por ellos mismos. Aun cuando hay cambios deberían coexistir versiones diferentes que permitan a los clientes cambiar sobre el tiempo. Esto permite optimizar la velocidad de nuev

Principios	Descripción
	as características e incrementar la autonomía de los equipos ya que no necesitan orquestar sus despliegues. Se puede adoptar por el modelo un servicio por host que reduce el impacto de afectar a otro servicio no relacionado. En general se recomienda no que el despliegue de un servicio no bloquee a otro y que los clientes decidan cuando se actualizan ellos mismos.
Aislar las fallas	Una arquitectura de microservicios puede ser más resistente que un sistema monolítico, pero solo si se cuenta con un plan de fallas dentro del sistema. Si no se cuenta con el hecho de que puede y va a fallar, el sistema va a comenzar a fallar y puede hacer al sistema más frágil. Cuando se usan llamadas no se debe tratar las llamadas remotas como las llamadas locales en las que hay una especie de fallo.
	Comprender que y como usar los circuit breakers (interruptores automáticos) para limitar las consecuencias de un componente defectuoso. Comprender el impacto de que pasaría si una parte del sistema se comporta erróneamente, a veces se debe sacrificar la disponibilidad o consistencia.
Altamente Observables	No se puede confiar en observar el comportamiento de una sola interfaz para ver si el sistema funciona correctamente, Debe usarse una vista holística. Usar monitoreo con transacciones de prueba que simulen un comportamiento real. Es importante agregar logs, estadísticas y usar IDS correlacionados que permitan el rastreo de las llamadas del sistema.

Fuente: Adaptado de (Newman, 2015, pp. 45-50).

Al aplicar la arquitectura de microservicios, es importante tener en cuenta las ventajas que acarrea estos, según Newman, (2015, pp. 4-7).

Tabla 5
Ventajas de la arquitectura de microservicios

Ventajas	Descripción
Tecnología Heterogénea	Dado que los microservicios pueden estar en diferentes sitios independientes unos de otros se pueden usar diversidades de tecnología, esto ayuda a encontrar la herramienta apropiada para el tipo de trabajo que se realice en vez de contar una sola tecnología estandarizada. Con los microservicios se logra que nuevas tecnologías puedan ser integradas rápidamente, mayormente a que los riesgos son mucho menores que una sola aplicación monolítica.

Ventajas	Descripción
Resistencia	La arquitectura de microservicios debe ser resistente a fallos. Si algo falla y no están en cascada se puede aislar el problema y el aplicativo continúa trabajando; por otro lado, en la arquitectura monolítica si el servicio falla todo el aplicativo debe pararse, una forma de mitigar el error se corría en varias máquinas el aplicativo. Sin embargo, el autor menciona que se debe ser cuidadoso para asegurar que los microservicios puedan mejorar la resistencia a fallos. La red puede y va a fallar para eso la arquitectura debe estar preparada.
Escalabilidad	Cuando se tiene un servicio monolítico todo debe ser escalado junto, dado que está relacionado y restringido por lo grande que es el aplicativo. Con más pequeños servicios se puede escalar solo los servicios que se desean
Facilidad de desplegar	En una aplicación monolítica la tarea de desplegar conlleva muchos riesgos, lanzar una versión demanda mucho tiempo, ya que se toma un tiempo para solucionar todos los posibles errores entre versiones. Con microservicios se pueden realizar cambios y desplegar solo dicho servicio, esto permite nuestro código desplegar rápidamente, lo cual permite que las nuevas funcionalidades se puedan ver más rápido
Alineamiento con la organización	Existen problemas asociados con largos equipos y largos códigos fuentes. Es realidad que los equipos más pequeños con códigos fuentes más pequeños tienden a ser más productivos. Microservicios ayuda a alinear la arquitectura a la organización, ayudando a reducir el número de personas trabajando sobre un código fuente.

Fuente: Adaptado de (Newman, 2015; pp. 4-7).

Pero también se debe considerar las desventajas, que según Fowler & Lewis, (s.f.), se pueden encontrar en el uso de la arquitectura microservicios.

Tabla 6
Desventajas de la arquitectura de microservicios

Desventajas	Descripción
Distribución	La complejidad de realizarse esta arquitectura de forma distribuida es: 1) Rendimiento: Es realizar llamadas a funciones remotas, las cuales son lentas a comparación de llamadas a funciones en el proceso. Esto se puede mitigar mediante: Incremento de granularidad (menos

Desventajas	Descripción
	llamadas de funciones remotas asincrónicamente). 2) Confiabilidad: Las llamadas a funciones pueden fallar, si hay más cantidad de microservicios la probabilidad de que ocurra es más alta. Esto se puede mitigar mediante la colaboración asincrónica ajustando el manejo de fallos y pueda resultar en una mejor recuperación.
Consistencia eventual	Con los microservicios la cantidad de consistencias eventuales va a crecer debido a que la base de datos esta descentralizada, mientras que en una aplicación monolítica todo forma parte de una transacción. Para esto los desarrolladores deben prever estos problemas de la consistencia y saber cómo detectarlos antes de que el código ocasione estos problemas.
Complejidad operacional	Cada unidad es independiente para ser desplegada; pero esto añade otro tipo de complejidades como el tener un número mucho mayor de microservicios que de aplicaciones. Manejar esta complejidad requerirá nuevas habilidades y herramientas como la entrega continua o la monitorización de los microservicios; aun así, seguirá siendo más compleja que una arquitectura monolítica. Además, se necesita introducir la cultura "devops" que consiste en una excelente colaboración entre desarrolladores, operaciones y todo aquel involucrado en la entrega del software.

Fuente: Adaptado de (Fowler & Lewis (s.f.)).

Para decidir que arquitectura es la más idónea en estos casos se debe hacer un análisis de benchmarking, por lo que en la Tabla 7, se presenta una comparación entre los microservicios y SOA, lo cual permitió en esta investigación decidir utilizar la metodología descrita.

Tabla 7
Comparativa entre microservicios y SOA

Asuntos	Microservicios	SOA
Despliegue	Despliegue individual del servicio	Despliegue monolítico todos a la vez
Equipos	Cada microservicio es manejado por equipos individuales	Servicios, integración y la interfaz de usuario son manejados por equipos individuales.

Asuntos	Microservicios	SOA
Interfaz de usuario	Parte de un microservicio	Portal de todos los microservicios
Alcance de la arquitectura	Un proyecto	Toda la compañía/empresa
Flexibilidad	Despliegue rápido e independiente	Ajusto de procesos de negocios en el límite de los servicios
Mecanismo de integración	Integración simple y primitiva	Integración inteligente y compleja
Nube nativa	Si	No
Administración / Gobierno	Distribuido	Centralizado
Almacenamiento de datos	Por unidad	Compartida
Escalabilidad	Mejor escalabilidad horizontal. Elástico	Limitado comparado a microservicios, cuello de botella en integrar unidades o sobrecarga de uso de análisis de mensajes. Elasticidad limitada
Unidad	Autónomo, desacoplado, propio contenedor, escalable independientemente.	Base de datos compartida, enlaces de unidades para servir a procesos de negocios. Bajo acoplamiento
Comunicación principal	Coreografía	Orquestación
Encaja	Infraestructuras de tamaño mediano	Infraestructuras de tamaño largo
Tamaño de servicio	Fina granularidad, pequeño	Granularidad fina o gruesa
Versionamiento	Debería ser parte de la arquitectura, más abierta a cambios	Mantenimiento de múltiples servicios de diferentes versiones
Nivel de Administración	Anarquía	Centralizado
Localización de las reglas de negocio	Servicio particular	Componente de integración

Fuente: Cerny, Donahoo, & Pechanec, (2017).

Con base en las comparaciones mostradas se puede resaltar en términos generales que en este estudio se eligió la arquitectura de microservicios dado que se ajusta más a las necesidades del cliente.

te, debido a que el software se desplegará en una empresa de nivel intermedio como es la EPO, aplicando el escalamiento horizontal, ya que es independiente no requiere de integraciones complejas como buses, tiene componentes con bajo acoplamiento, es flexible por contar con despliegue rápido que no afecta para el despliegue de otros servicios, el gobierno descentralizado de cada servicio contiene la lógica de negocio y su acceso a la información está aislado de otros servicios. Por último, su adaptación es más rápida para los cambios solicitados por los clientes.

DESARROLLO DE LA PROPUESTA

El alcance en este artículo cubrirá el análisis, planificación y diseño.

Análisis y Planificación.

Según las características del escenario (Britto, 2016), se optó por la metodología Scrum. Siguiendo sus pasos se conformó el equipo para el desarrollo de la propuesta descrita, Figura 3.

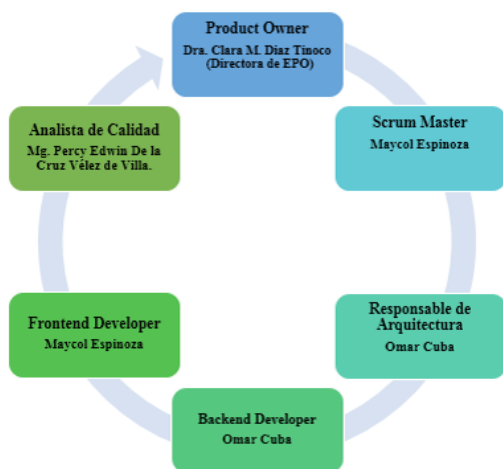


Figura 3.
Equipo Scrum del estudio (2019).

Planificación de los sprints (Inceptions o sprint 0)

En esta etapa participan el Product Owner, Scrum Master e incluimos al responsable de la Arquitec-

tura. Generándose el Backlog a partir de las épicas que el proyecto necesitaba. Se identifican las historias de usuario, y el Product Owner prioriza e identifica los hitos a controlar y la estimación del tiempo del proyecto. Se tiene proyectado a 6 sprints (0-5), cada sprint tendrá una duración de un mes. Es decir, la duración estimada del proyecto es de 6 meses. Además, se especifica la arquitectura de proyecto, la figura 4 detalla la misma.

En la Tabla 8, se ha integrado el Backlog general con una estimación referencial.

Tabla 8
BACKLOG

BAC-KLOG	Historias de Usuario	Estimación
HU01	Como usuario, debo poder autenticarme en el sistema de manera segura, así como garantizar el cierre de sesión en mi dispositivo móvil.	5
HU02	Como usuario, debo tener acceso a mis funcionalidades básicas correspondientes.	3
HU03	Como usuario, debo poder visualizar la relación de cursos, así como la estructura curricular de cada una de ellas.	4
HU04	Como usuario debo poder distinguir los estados de las clases correspondientes a un sílabo determinado.	3
HU05	Como docente debo ser capaz de iniciar y finalizar la clase, por consiguiente, establecer su estado como iniciado, finalizado respectivamente.	5
HU06	Como usuario deseo ser capaz de visualizar el detalle de la asistencia de una clase dada.	5
HU07	Como docente, debo poder consultar las calificaciones de todos mis alumnos inscritos en una determinada agrupación curricular	3
HU08	Como alumno, debo poder consultar mis calificaciones relacionadas a una agrupación curricular determinada.	3
HU09	Como docente, debo poder registrar o modificar una o más de las calificaciones de los alumnos inscritos en una agrupación curricular determinada	5
HU10	Como usuario, debo ser capaz de visualizar el listado de anuncios realizados por el docente a lo largo del semestre académico.	3
HU11	Como docente, debo poder registrar un anuncio correspondiente a una agrupación curricular determinada.	3

BAC-KLOG	Historias de Usuario	Estimación
HU12	Como usuario debo poder ver todas las preguntas y respuestas relacionadas a una clase determinada.	5
HU13	Como usuario, debo ser capaz de realizar preguntas y/o responder otras preguntas.	5
HU14	Como usuario, debo poder visualizar la formula correspondiente a una agrupación curricular.	5
HU15	Como docente, debo ser capaz de modificar una formula previamente definida correspondiente a una agrupación curricular.	3
HU16	Como alumno, debo valorizar la clase impartida por el docente.	3
HU17	Como usuario, debo ser capaz de ser notificado cuando ocurra algún evento relevante.	5
HU18	Como director de escuela debo ser capaz de ver los reportes de asistencias	5
HU19	Como director de escuela debo ser capaz de ver el reporte de desempeño académico estudiantil	5
HU20	Como director de escuela debo ser capaz de ver el reporte de valoraciones obtenidas en las encuestas	5
SPI-KE01	Diseño de la arquitectura Backend	8
SPI-KE02	Diseño de la arquitectura Frontend	8
SPI-KE03	Abordar la consistencia eventual en la arquitectura de microservicios	5
SPI-KE04	Implementación de Websockets en React Native	5

Fuente: Cerny, Donahoo, & Pechanec, (2017).

Definición de sprint

En los sprints del 1 al 5, la duración aproximada es de 1 mes por sprint y la jornada de trabajo es de lunes a viernes de 9am a 5pm (considerar hora refrigerio). El primer día del sprint comienza con una reunión de planificación de 4 a 7 horas. Aquí se estima la complejidad de la historia para entregar, asimismo se realiza el desglose de las tareas por persona y se asigna un responsable de la historia respectiva. Los siguientes días se realiza una reunión diaria de 15 minutos con el fin de revisar el avance, dificultades que tuviera el equipo. Posteriormente se realizará las tareas pactadas

en la planificación. En la tercera semana hay una reunión de refinamiento donde se presenta las historias que vendrán en el siguiente sprint (Se analizan riesgos y problemas etc.). Al final del sprint se realiza la revisión con la dueña del producto en un tiempo estimado de 3 horas (presentar una demostración).

El último día del sprint se realiza la reunión de retrospectiva del equipo, con una duración de 3 horas (para una mejorar continua del equipo).

Se describen en las Tablas 9, 10 y 11, algunos de los sprints de la propuesta que contemplan mayor valor para el usuario final (Control de asistencias y reportes).

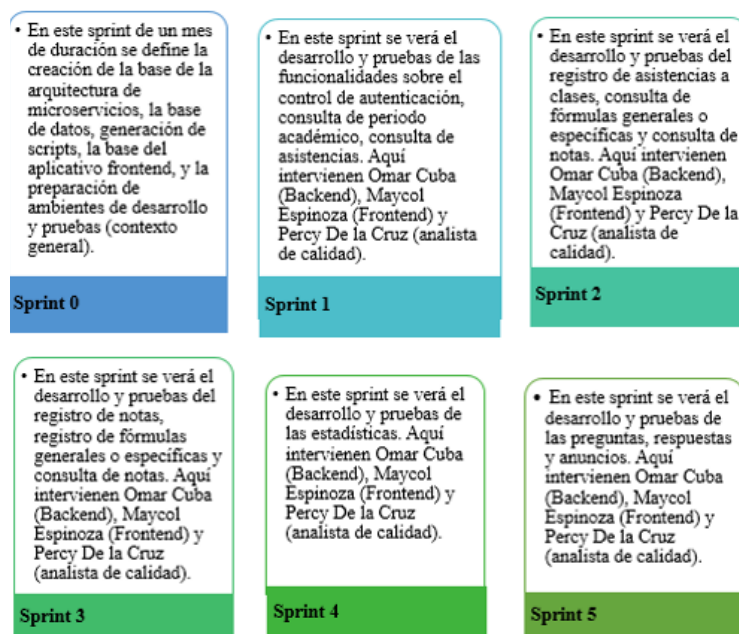


Figura 4. Descripción del Sprint 0 – 5. Elaboración propia (2019).

Tabla 9
 Sprint 1

HU / Spike	Tareas	Responsable
HU04	Creación del endpoint, Listar asistencia clases por agrupación curricular (MS ASISTENCIA)	Omar Cuba
	Creación del endpoint composer Listar Clases del presente día con sus respectivos estados (MS Periodo Académico + MS Asistencia)	Omar Cuba

HU / Spike	Tareas	Responsable
	Desarrollo del semáforo (estado de la clase) en la GUI Listar Clases del presente día	Maycol Espinoza
	Desarrollo del semáforo (estado de la clase) en la GUI Listar Clases por fecha específica	Maycol Espinoza
	Creación del esquema, tablas, constraints, queries correspondientes a la base de datos de ASISTENCIA.	Maycol Espinoza
	Creación y ejecución de los casos de pruebas alineados a los criterios de aceptación establecidos	Percy De la Cruz

Fuente: Elaboración propia, (2019).

Tabla 10
Sprint 2

HU / Spike	Tareas	Responsable
HU05	Creación del endpoint transaccional Iniciar Asistencia (MS ASISTENCIA)	Omar Cuba
	Creación del endpoint transaccional Finalizar Asistencia (MS ASISTENCIA)	Omar Cuba
	Creación del endpoint transaccional Registrar Asistencias (MS ASISTENCIA)	Omar Cuba
	Creación del endpoint Listar Alumnos por agrupación curricular (MS PERIODO ACADEMICO)	Omar Cuba
	Integración de los endpoints mencionados anteriormente dentro de la capa Frontend.	Maycol Espinoza
	Implementación del registro de asistencias de los alumnos en la GUI Asistencia.	Maycol Espinoza
	Implementación del inicio y finalización de la clase en la GUI Asistencia	Maycol Espinoza
	Creación y ejecución de los casos de pruebas alineados a los criterios de aceptación establecidos.	Percy De la Cruz
HU06	Adaptación del componente acordeón en el GUI Asistencias para añadir el detalle de asistencias	Maycol Espinoza
	Asegurar el comportamiento responsivo para ambas partes del acordeón (Detalle de clase y Listado de asistencias)	Maycol Espinoza

HU / Spike	Tareas	Responsable
	Implementar las "go backs" (regresar atrás Android) en todas las vistas realizadas hasta el momento.	Maycol Espinoza
	Creación y ejecución de los casos de pruebas alineados a los criterios de aceptación establecidos	Percy De la Cruz

Fuente: Elaboración propia, (2019).

Tabla 11
Sprint 4

HU / Spike	Tareas	Responsable
HU18	Creación del endpoint "obtenerEstadisticasAsistencias" (MS ASISTENCIA), servicio encargado de obtener los porcentajes de faltas, tardanzas, asistencias.	Omar Cuba
	Creación de la GUI "Ver reportes de asistencias", con sus respectivos filtros: "curso", "agrupación curricular", "docente" y "alumno"	Maycol Espinoza
	Creación y ejecución de los casos de pruebas alineados a los criterios de aceptación establecidos	Percy De la Cruz
HU19	Creación del endpoint "obtenerEstadisticasNotas" (MS NOTAS), servicio encargado de obtener los porcentajes según rango de notas. Del mismo modo, con los filtros: "curso", "agrupación curricular", "estudiante".	Omar Cuba
	Creación del GUI "Ver reportes de Desempeño Académico", con sus respectivos filtros: "curso", "agrupación curricular", "estudiante".	Maycol Espinoza
	Creación y ejecución de los casos de pruebas alineados a los criterios de aceptación establecidos	Percy De la Cruz
H20	Creación del endpoint "obtenerEstadisticasEncuestas", servicio encargado de obtener el promedio de valoraciones respecto a un curso y/o agrupación curricular seleccionados.	Omar Cuba
	Creación del GUI "Ver reporte de encuestas", con sus respectivos filtros: "curso" y "agrupación curricular"	Maycol Espinoza
	Creación y ejecución de los casos de pruebas alineados a los criterios de aceptación establecidos	Percy De la Cruz

HU / Spike	Tareas	Responsable
	Creación del GUI "Ver comparativo Notas vs Encuestas", con sus respectivos filtros "Curso" y "agrupación curricular".	Maycol Espinoza
	Creación y ejecución de los casos de pruebas alineados a los criterios de aceptación establecidos	Percy De la Cruz

Fuente: Elaboración propia, (2019).

Diseño

En la figura 5, se describe el diseño de la arquitectura del sistema que soportará la futura implementación del proyecto. Se considera su composición de la siguiente forma: Por el lado cliente (frontend) Eisenman (2016) afirma que usar el framework react native permite la renderización en aplicativos con plataformas móviles como ANDROID e IOS. Por el lado del servidor (backend) se cuentan con los componentes no funcionales que apoyan a nuestra arquitectura de microservicios (config, eureka, gateway y composición), y los componentes funciones, los cuales son los microservicios de autenticación, asistencias, notas, coparticipación, encuesta, notificación, reportes que serán desarrollados con spring boot y cada uno contará con un acceso privado a su base de datos. Para agregar seguridad de acceso a los servicios se hará uso de oauth2 y por último para gestionar los eventos (mediante mensajería) se usará rabbitmq.

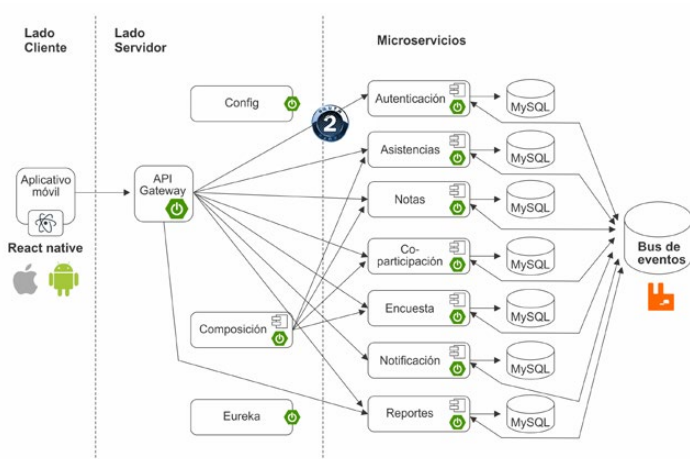


Figura 5.
 Diagrama del diseño de la arquitectura del sistema.
 Fuente: Elaboración propia (2019).

A continuación, se detallará cada uno de los componentes funcionales en la Tabla 12 y los componentes no funcionales en la Tabla 13 de la arquitectura de microservicios propuesta.

Tabla 12
 Descripción de los componentes no funcionales

Componentes no funcionales	Elementos no funcionales
No contienen lógica de negocio, pero aportan en el funcionamiento de la arquitectura de microservicios se hace uso del framework spring cloud y también se considera aquí al servicio composición el cual requiere el uso de la combinación de varios microservicios para las consultas del lado cliente (frontend).	Eureka-Server. Este servicio usa el componente spring cloud eureka nos permite realizar el auto-registro, descubrimiento dinámico y balanceo de carga. (RV, 2016)
	Config-Server. Este servicio usa el componente spring cloud config el cual usamos la configuración externa de las propiedades de configuración de los microservicios.
	Gateway. Este servicio se sitúa entre el cliente y el servidor, transforma la data según lo requerido por el cliente (servicios UI). Actúa como un proxy para el backend, exponiendo un conjunto de APIs específicas al cliente. Haremos uso de zuul, que usa el proxy reverso que funciona como una plataforma compartida para varios microservicios.
	Composición. Este servicio usa el patrón API composition, este patrón implementa una consulta(query) al invocar a los servicios que tienen la data por medio de sus APIs y las combina los resultados en un resultado. No cuenta con base de datos (BD).

Fuente: Elaboración propia (2019).

Tabla 13

Descripción de los componentes funcionales

Componentes funcionales	Elementos funcionales
<p>Aquí se encuentra la lógica de negocio, aquí están los microservicios. Cada uno es independiente entre sí definidos así por límites de negocio, manejan su propia base de datos. Para la seguridad se hace uso de Oauth2 que permite acceso a los servicios cuando el usuario se autentica mediante tokens. Por último, se hace uso de rabbit MQ para que los microservicios al registrar cambios se comuniquen entre sí mediante eventos (mensajes). Esto permite la programación asincrónica y el uso de colas en cambio la comunicación mediante HTTP por defecto no es secuencial es decir aumenta el tiempo de ejecución del servicio y no es recomendable a menos que se maneje de forma asincrónica.</p>	<p>Autenticación. Este microservicio contiene la información relacionada al usuario, roles y perfiles; permite a un usuario con perfiles como (profesor, alumno, directora de escuela) la autenticación en el sistema mediante el ingreso de su usuario y contraseña; también permite realizar el cierre de sesión de un usuario dado. Aquí se usa Oauth2 el cual brinda token al cliente que permitirá el acceso a los microservicios. Cuenta con una BD mysql.</p>
	<p>Asistencia. Permite registrar/actualizar y finalizar las asistencias de las clases, Cuenta con una BD mysql.</p>
	<p>Encuesta. Permite realizar las encuestas con calificaciones serán de 1 a 5 estrellas. Permite obtener la última plantilla de encuesta en estado activo con la cual se realizarán las encuestas. Por último, permite calcular el promedio de encuestas realizadas en base a las clases. Cuenta con una BD mysql.</p>
	<p>Coparticipación. Entre las principales funciones de este microservicio serán de "registrar pregunta", referentes a un tema en específico o en general para que puedan ser respondidas por el docente o en todo caso por un compañero de clase. "responder respuesta" es el caso consecuente de la acción "registrar pregunta", puede ser realizado por profesores y/o alumnos con el fin de solventar la duda. "Registrar anuncios (por parte de los profesores)", que permite al profesor hacer anuncios de algunos imprevistos o avisos que quiera comunicar. Cuenta con BD mysql.</p>
	<p>Notas. Los docentes asignados a los grupos de un curso podrán ingresar las notas de las diferentes evaluaciones. etc.</p>

Componentes funcionales	Elementos funcionales
	<p>Así como la consulta y el registro de fórmulas generales y específicas. Cuenta con una BD mysql.</p>
	<p>Periodo Académico. Contiene la información de los cursos en un semestre, tiene la información de los alumnos y profesores, los grupos definidos en el curso, la relación de alumnos inscritos en dichos grupos y también la relación de los horarios de las clases para cada grupo. Cuenta con una BD mysql.</p>
	<p>Notificación Es útil cuando hay cambios o actualizaciones por parte de los alumnos y profesores que es de importancia para los demás, se resaltan las notificaciones de asistencias, notas, anuncios, preguntas y respuestas. Cuenta con BD mysql.</p>
	<p>Reportes. Permitirá generar reportes y/o estadísticas. Cuenta con una BD propia para una consulta más rápida, la cual es llenada mediante eventos por mensajería usando rabbitMQ que usan los microservicios de asistencias, notas, encuestas y periodo académico cuando actualizan su propia BD. Cuenta con una BD mysql.</p>

Fuente: Elaboración propia (2019).

Perfiles. La propuesta desarrollada para la EPO, tiene 3 perfiles: alumno, profesor y de la directora de escuela. Las interfaces funcionales del menú de la directora de escuela, se aprecia en la figura 6.

Estadísticas: Le permite generar estadísticas en base a las asistencias, notas y encuestas de alumnos y profesores.

Cambiar Rol: Le permite cambiar de rol si tuviera más de uno.

Cerrar Sesión: Le permite cerrar la sesión

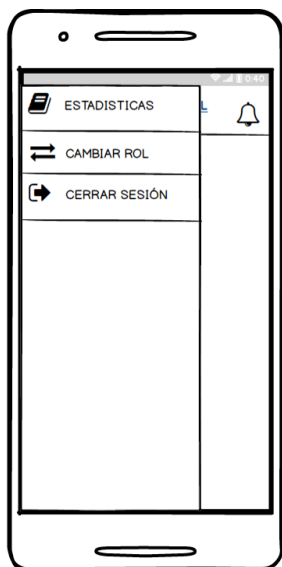


Figura 6.
 Menú de principales funciones de la directora de escuela.

Las interfaces funcionales del menú del profesor, se aprecia en la figura 7.

Asistencias Hoy: Le permite gestionar o consultar las asistencias del día de fecha actual de las clases de sus cursos.

Asistencias: Le permite gestionar o consultar las asistencias de todas las clases de sus cursos.

Notas: Le permite gestionar o consultar las notas de los alumnos de sus cursos.

Preguntas y Respuestas: Le permite registrar o consultar preguntas y respuestas con los alumnos.

Anuncios: Le permite registrar o consultar anuncios de sus cursos.

Fórmula: Le permite gestionar o consultar fórmulas del cálculo de las notas de sus cursos.

Estadísticas: Le permite generar estadísticas sobre las asistencias, notas y encuestas de alumnos y profesores.

Cambiar Rol: Le permite cambiar de rol si tuviera más de uno.

Cerrar Sesión: Le permite cerrar la sesión

Las interfaces funcionales del menú del alumno, se aprecia en la figura 8.

Asistencias Hoy: Le permite consultar las asistencias del día de fecha actual de las clases de sus cursos.

Asistencias: Le permite consultar las asistencias de todas las clases de sus cursos.

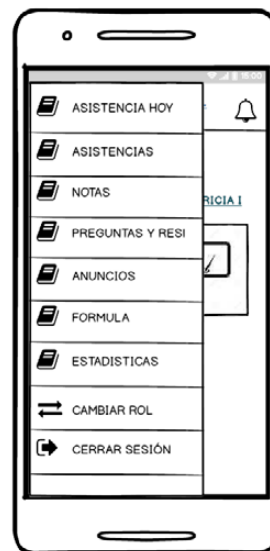


Figura 7.
 Menú de principales funciones del profesor.

Notas: Le permite consultar las notas de sus cursos.

Preguntas y Respuestas: Le permite registrar o consultar preguntas y respuestas con los profesores y/o alumnos.

Anuncios: Le permite consultar los anuncios del profesor sobre sus cursos.

Fórmula: Le permite consultar fórmulas del cálculo de las notas de sus cursos.

Cambiar Rol: Le permite cambiar de rol si tuviera más de uno.

Cerrar Sesión: Le permite cerrar la sesión

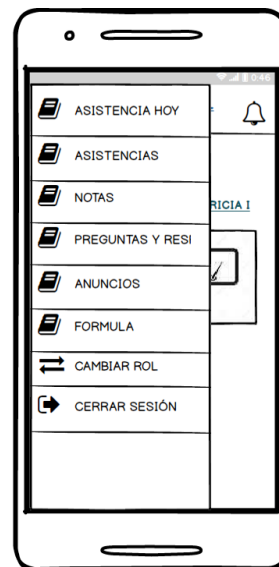


Figura 8.
 Menú de principales funciones del alumno.

Una de las funcionalidades más importantes del sistema se visualiza en la figura 9. Permitirá realizar el control general y específico de asistencias de alumnos y profesores por la directora de escuela. Los filtros que se presentan para lograr este fin son el curso, grupo (todos o uno específico), alumno (todos o uno específico), profesores (todos o uno específico), luego de haber escogido los criterios de filtros deseados se muestran estadísticas (que permitirán interpretar mejor la información) como avance de clases (porcentaje de clases avanzadas comparado con el total de clases) y dos gráficos por el momento de asistencias: el primero de profesores y el segundo de alumnos.

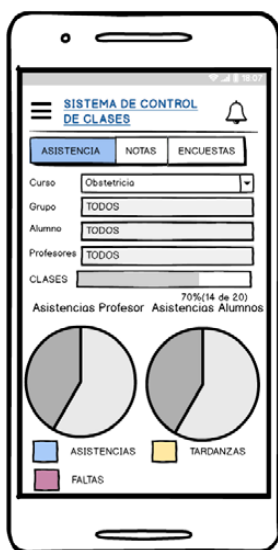


Figura 9. Reporte/Consulta para control de asistencias de alumnos y profesores.
Fuente: Elaboración propia (2019).

CONCLUSIONES

Se ha presentado una propuesta de diseño de un sistema móvil de control académico basado en microservicios y desarrollado con la metodología Scrum para la Escuela Profesional de Obstetricia (EPO) de la Universidad Nacional Mayor de San Marcos, en la se muestran las funcionalidades que debería tener el producto cuando se implemente, satisfaciendo las necesidades de información e integración. La arquitectura de microservicios presenta independencia, acoplamiento, cohesión, flexibilidad, despliegue y descubrimiento de servi-

cios, por otro lado, la metodología Scrum permite la participación activa de los usuarios y mejor control para el logro de los entregables, así como una mejor gestión y organización del proyecto, facilitando el desarrollo y sobre todo permitir culminar el proyecto en los plazos establecidos.

Este artículo describe la parte de planificación, análisis y diseño del avance de una tesis de Ingeniería que aún está por concluirse, para su implementación previamente habría que probar los despliegues, así como una mejor distribución de los microservicios alojados en la base de datos, según tuning a realizar podría replantearse y refinarse la arquitectura presentada en la figura 5.

El trabajo futuro a realizar sería continuar con las fases de desarrollo e implementación, la cual permitirá brindar una herramienta a la EPO y similares; facilitando la gestión académica y cumplir con los estándares planteados y exigidos por SINEACE.

REFERENCIAS BIBLIOGRÁFICAS

- Abernethy, M.A. & Chua, W.F. (1996). Un estudio de campo del "rediseño" del sistema de control: el impacto de los procesos institucionales en la elección estratégica. *Contemporary Accounting Research*, 13 (2), 569-606. <https://doi.org/10.1111/j.1911-3846.1996.tb00515.x>
- Arroyo, M.N. (2009) Influencia de la gestión pedagógica en el uso de las tecnologías de la información y comunicación en la Institución Educativa Darío Arrús de Bellavista, Callao. (Tesis inédita de Maestría). Universidad Nacional "Enrique Guzmán y Valle. Lima.
- Beuren, I.M. & Teixeira, SA (2014). Evaluación de sistemas de control de gestión en una institución de educación superior con gestión y control del desempeño <https://doi.org/10.4301/S1807-17752014000100010>
- Britto, J.A. (2016). Comparación de metodologías ágiles y procesos de desarrollo de software mediante un instrumento basado en CMMI. *Scientia et Technica*. 21 (2), 150-155. <https://doi.org/10.22517/23447214.9249>
- Cerny, T., Donahoo, M. & Pechanec, J. (2017). Desambiguación y comparación de SOA, microservicios y sistemas autónomos. *Actas del RACS '17 de la Conferencia Internacional sobre Investigación en Sistemas Adaptativos y Convergentes* (pp. 228-235). NY, EE. UU: Asociación de Maquinaria Informática. <https://doi.org/10.1145/3129676.3129682>

- Che, Z. & Rapiyah, M. (2013). El efecto del sistema de control de gestión en el sistema de medición de desempeño en Small Medium Hotel en Malasia *Revista Internacional de Comercio, Economía y Finanzas*, 4 (4), 202-208. <https://doi.org/10.7763/IJTEF.2013.V4.286>
- COMEXPERU. (2011) Educación Superior: un diagnóstico. *Semanario COMEXPERU* (636), 5-6. Recuperado de <http://www.comexperu.org.pe/media/files/semanario/SEMANARIO%20COMEXPERU%20636.PDF>
- Eisenman, B. (2016). *Learning React Native* (Primera edición). (M. Foley, Ed.) Estados Unidos de América: O'Reilly Media Inc.
- Erl, T. (2016). *SOA: Principios del diseño del servicio*. NJ, EE. UU.: Prentice Hall.
- Fernández, J.M. & Cadelli, S. (2014). Convivencia de metodologías: Scrum y Rup en un proyecto de gran escala (Tesina de Licenciatura). Universidad Nacional La Plata, Argentina. Recuperado de <http://hdl.handle.net/10915/47082>
- Ferreira, A. & Otley, D. (2006). El diseño y uso de sistemas de gestión del desempeño: un marco extendido para el análisis. *Investigación de contabilidad de gestión*, 20 (4), 263-282. <https://doi.org/10.1016/j.mar.2009.07.003>
- Fowler, M. & Lewis, J. (s.f). *Guía de recursos de microservicios*. Recuperado de <https://martinfowler.com/microservices/>
- Haimann, T. (2005). En *Dirección y gerencia: planificación, coordinación y control de las actividades de la empresa*.
- Horngren, C.T., Datar, S.M. & Foster, G. (2000). *Contabilidad de costos*. Río de Janeiro: LTC.
- Krafzig, D., Banke, K. & Slama, D. (2004). *Enterprise SOA: mejores prácticas de arquitectura orientada a servicios*. Prentice Hall.
- Moromi, NH (2002). La Influencia de la ejecución curricular y el uso de medios y materiales en el rendimiento académico de los estudiantes de la Facultad de Odontología de la Universidad Nacional Mayor de San Marcos (Tesis Postgrado). Universidad Nacional Mayor de San Marcos, Escuela de Postgrado - Unidad de Posgrado de la Facultad de Educación. Lima: Universidad Nacional Mayor de San Marcos. Recuperado de http://sisbib.unmsm.edu.pe/BibVirtual/Tesis/Human/Moromi_N_H/Moromi_N_H.htm
- Newman, S. (2015). *Construcción de microservicios*. (ML MacDonald, Ed.) O'Reilly.
- Pérez-Herrera, M. (2015). *Arquitecturas basadas en microservicios*. Madrid, España: UPM.
- Quacquarelli, S. (2011). *QS University Rankings: Latin America 2011*. Recuperado el de https://content.qs.com/supplement2011/Latin_American_supplement.pdf
- Quacquarelli, S. (2019). *QS University Rankings: Latin America 2019*. Quacquarelli Symonds: <https://www.topuniversities.com/university-rankings/latin-american-university-rankings/2019>
- Rodríguez, C. & Herrera, L. (2009). Análisis correlacional-predictivo de la influencia de la asistencia a clase en el rendimiento académico universitario. Estudio de caso en una asignatura. *Profesorado: Revista de curriculum y formación del profesorado*, 13 (2), 1-13. Recuperado de <http://www.redalyc.org/articulo.oa?id=56711798017>
- RV, R. (2016). *Spring Microservices*. Packt Publishing.
- Sánchez Enriquez, JA (2011). Control de obtención del rendimiento de recursos humanos. (Thomson Reuters, Ed.).
- Sifuentes, L.A. (2018). Desempeño didáctico y académico del docente relacionado con la satisfacción de los estudiantes del programa de Complementación Pedagógica de la Universidad Nacional Mayor de San Marcos, 2013-II. Recuperado de <http://cybertesis.unmsm.edu.pe/handle/cybertesis/3981>
- SINEACE (2016) *Modelo de Acreditación para Programas de Estudios de Educación Superior Universitaria*. Lima, Perú. Recuperado de <https://www.sineace.gob.pe/wp-content/uploads/2016/03/Anexo-a-la-Resoluci%C3%B3n-N%C2%B0022-2016-CDAH-P.pdf>
- Tolentino, L.A. (2014). Desempeño didáctico y académico del docente relacionado con la satisfacción de los estudiantes del programa de Complementación Pedagógica de la Universidad Nacional Mayor de San Marcos, 2013-II (Tesis Postgrado). Universidad Nacional Mayor de San Marcos, Facultad de Educación. Lima-Perú: Universidad Nacional Mayor de San Marcos. Recuperado de <http://cybertesis.unmsm.edu.pe/handle/cybertesis/3981>
- Yanada, G., Rivera, M. & Castro, J. (2012). *Educación Superior en el Perú: Retos para el Aseguramiento de la Calidad*. Perú: SINEACE. Recuperado de <http://repositorio.minedu.gob.pe/handle/123456789/937>