



**FACULTAD DE INGENIERIA, ARQUITECTURA Y
URBANISMO**

ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS

TESIS

**METODOLOGÍA DE CONVERSIÓN DE
APLICACIONES MONOLÍTICAS A
MICROSERVICIOS DESPLEGABLE EN LA NUBE
PARA PEQUEÑAS EMPRESAS**

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

Autor:

**Bach. Mendoza Linares Jersson German
ORCID: <http://orcid.org/0000-0003-1143-8158>**

Asesor:

**Mg. Atalaya Urrutia Carlos William
ORCID: <http://orcid.org/0000-0002-2761-4868>**

Línea de Investigación:

Infraestructura, Tecnología Y Medio Ambiente

Pimentel – Perú

2022

APROBACIÓN DEL JURADO

METODOLOGÍA DE CONVERSIÓN DE APLICACIONES MONOLÍTICAS A MICROSERVICIOS DESPLEGABLE EN LA NUBE PARA PEQUEÑAS EMPRESAS

Bach. Mendoza Linares Jersson German
Autor

Mg. Atalaya Urrutia Carlos William
Asesor

Mg. Bravo Ruiz Jaime Arturo
Presidente de Jurado

Mg. Atalaya Urrutia Carlos William
Secretario de Jurado

Mg. Diaz Vidarte Miguel Orlando
Vocal de Jurado

DEDICATORIA

Dedico este trabajo a mi familia.

A mi padre Segundo Germán Mendoza Baldera, por darme la oportunidad de ser alguien en la vida y ser mi guía desde el cielo.

A mi madre Ana Rosa Linares Diaz, por siempre apoyarme y darme fuerzas en el proceso de formación académica y toda mi vida.

A mi hermano Alexis Ansony Mendoza Linares, por brindarme su apoyo y confiar en mi persona.

A mi abuelita Florentina Diaz Loayza, por brindarme siempre sus consejos, sabiduría, y apoyo.

A mi tía Lucia Linares Diaz, por brindarme su apoyo constante.

AGRADECIMIENTO

“El sabio no acumula,
obra enteramente para los otros
y posee cada vez más.
Lo da todo a los demás,
y tiene más cada vez”

Lao Tse

Agradecer a Dios y mi Padre, por guiarme desde el cielo por el camino del bien, a mi familia por su constantemente apoyo en el proceso de mi formación universitaria, a mi grupo de amigos por creer siempre en mí, su amistad y apoyo, a los docentes por brindarme su apoyo, amistad y conocimiento durante mi formación académica y a la Universidad Señor de Sipán por ser la sede de todo el conocimiento adquirido en estos años.

RESUMEN

Tradicionalmente el desarrollo de software estuvo enfocado en la arquitectura monolítica, dicha arquitectura acopla todo su proceso agrupando funcionalidades, ocasionando dependencias entre sí, obstáculos en escalabilidad, integración continua y sobrecargando todo el sistema en una sola base de código. Sin embargo, la arquitectura orientada a microservicios se desarrolló como propuesta para solventar los problemas ocasionados por la arquitectura monolítica, proporcionando granularidad, independencia y versatilidad en cuanto a la adopción de nuevas tecnologías, aportando a la integración, despliegue continuo, mantenimiento, productividad y desarrollo ágil. Empresas como Netflix, Amazon, Spotify empezaron con una aplicación monolítica, pero su crecimiento impidió la escalabilidad y las mejoras constantes, por ello migraron a la arquitectura de microservicios aprovechando los beneficios para el negocio. Sin embargo, realizar dicha migración presenta un problema de complejidad afectando el proceso de migración y desarrollo del software. En los últimos años las pequeñas empresas han empezado a migrar a la nube aprovechando los beneficios como la agilidad, elasticidad, disponibilidad, a ello se le suma la arquitectura de microservicios. Para realizar dicha migración se debe seguir una metodología de conversión que disminuya la complejidad y facilite el entendimiento. Por lo tanto, este proyecto de investigación permite el desarrollo de una metodología para la conversión de una aplicación monolítica a una basada en la arquitectura a microservicios a través de cinco actividades tales como el “Análisis de factibilidad de migración de arquitectura de software”, “Planeamiento de Software”, “Diseño de Software”, “Desarrollo de Software”, “Despliegue de Software”, esto permite ofrecer aportes sobre la arquitectura de microservicios que sirva como base en los diferentes modelos o proyectos que se vayan a realizar. La metodología se aplicó a un caso de estudio de una aplicación monolítica “GMO”, obteniendo un bajo grado de acoplamiento y alta cohesión entre los microservicios. De la misma manera se obtuvo una alta satisfacción, una baja complejidad y un aumento del 20,2% de la productividad de los integrantes involucrados en la aplicación del caso de estudio.

Palabras Claves — Arquitectura de microservicios, Aplicaciones monolíticas, Metodología.

ABSTRACT

Traditionally, software development was focused on monolithic architecture, said architecture couples its entire process by grouping functionalities, causing dependencies among themselves, obstacles in scalability, continuous integration and overloading the entire system in a single code base. However, the microservices-oriented architecture was developed as a proposal to solve the problems caused by monolithic architecture, providing granularity, independence and versatility in terms of the adoption of new technologies, contributing to integration, continuous deployment, maintenance, productivity and development. agile. Companies like Netflix, Amazon, Spotify started with a monolithic application, but its growth prevented scalability and constant improvements, so they migrated to the microservices architecture, taking advantage of the benefits for the business. However, performing such migration presents a complexity problem affecting the migration and software development process. In recent years, small companies have begun to migrate to the cloud, taking advantage of the benefits such as agility, elasticity, availability, in addition to the microservices architecture. To carry out this migration, a conversion methodology must be followed that reduces complexity and facilitates understanding. Therefore, this research project allows the development of a methodology for the conversion of a monolithic application to one based on the architecture to microservices through five activities such as the "Software architecture migration feasibility analysis", "Software Planning", "Software Design", "Software Development", "Software Deployment", this allows us to offer contributions on the microservices architecture that will serve as a basis for the different models or projects to be carried out. The methodology was applied to a case study of a monolithic "GMO" application, obtaining a low degree of coupling and high cohesion between the microservices. In the same way, high satisfaction, low complexity and a 20.2% increase in the productivity of the members involved in the application of the case study were obtained.

Keywords — Microservices architecture, Monolithic applications, Methodology.

ÍNDICE

I. INTRODUCCIÓN.....	8
1.1. REALIDAD PROBLEMÁTICA.....	10
1.2. TRABAJOS PREVIOS.....	14
1.3. TEORÍAS RELACIONADAS AL TEMA.....	22
1.4. FORMULACIÓN DEL PROBLEMA.....	47
1.5. JUSTIFICACIÓN E IMPORTANCIA DEL ESTUDIO.....	47
1.6. HIPÓTESIS.....	48
1.7. OBJETIVOS.....	48
1.7.1. OBJETIVOS GENERALES.....	48
1.7.2. OBJETIVOS ESPECÍFICOS.....	49
II. MATERIAL Y MÉTODO.....	49
2.1. TIPO Y DISEÑO DE INVESTIGACIÓN.....	49
2.2. POBLACIÓN Y MUESTRA.....	50
2.3. VARIABLES, OPERACIONALIZACIÓN.....	52
2.4. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS, VALIDEZ Y CONFIABILIDAD.....	53
2.5. PROCEDIMIENTO DE ANÁLISIS DE DATOS.....	54
2.6. CRITERIOS ÉTICOS.....	57
2.7. CRITERIOS DE RIGOR CIENTÍFICO.....	58
III. RESULTADOS.....	59
3.1. RESULTADOS EN TABLAS Y FIGURAS.....	59
3.2. DISCUSIÓN DE RESULTADOS.....	74
3.3. APORTE PRÁCTICO.....	76
IV. CONCLUSIONES Y RECOMENDACIONES.....	205
4.1. CONCLUSIONES.....	205
4.2. RECOMENDACIONES.....	206
REFERENCIAS.....	207
ANEXOS.....	219

I. INTRODUCCIÓN

La computación en la nube, también llamada cloud computing, permite escalar e implementar aplicaciones a las empresas y estas puedan desplegar sus propias aplicaciones ya sea en Infraestructura como servicio (IaaS) o Plataforma como servicio (PaaS), cuando las empresas despliegan sus aplicaciones se enfrentan a retos tales como auto escala, continuo Entrega, despliegues en caliente, alta disponibilidad, dinámica. Monitoreo, entre otros (Villamizar et al., 2015).

El software desarrollado por las empresas tiende a buscar escalabilidad, disponibilidad, integración, mejora rápida y continua. La arquitectura orientada a microservicios solventa estas desventajas. El presente trabajo de investigación describe el estudio realizado sobre la arquitectura basada en microservicios (MSA).

Realizar una migración de una arquitectura monolítica a microservicios tiene un proceso en el cual surgen complejidades desde la concepción de los requerimientos hasta el despliegue continuo del software. Como ejemplo tenemos a grandes empresas que migraron de su aplicación monolítica a microservicios, por ejemplo, Netflix se desplegaba como un único fichero “.war” (representación de una arquitectura monolítica) en un servidor de aplicaciones java. En 2016 recibe más de dos mil millones de peticiones a su API desde el exterior. Hace años la página de Amazon tenía más de 150 programas implicados en la generación de su página de web. Amazon hace un deploy en 2016 cada 0.6 segundos. En 2016 Spotify cuenta con 600 desarrollos divididos en 90 equipos autónomos y multidisciplinares, cada uno de ellos cuidando su propio grupo de microservicios. En total tienen desplegados más de 800 (Java) (Javit, 2016). Estas empresas migraron de arquitectura y ahora se basan en microservicios para su correcta disponibilidad, como nos podemos percatar, los microservicios se basan en la abstracción de los requerimientos de manera independiente, esto quiere decir que cada microservicio es aislado, por lo que ofrece ciertas ventajas tales como el versionado, escalabilidad, integración,

desarrollo agile.

Según una investigación realizada por el Synergy Research Group, indica que durante los primeros 4 meses de la pandemia del 2020, los servicios tecnológicos de empresas como Amazon Web Services (AWS) teniendo un crecimiento del 32%, Microsoft Azure alcanzando el 18% y Google Cloud quien están superando sustancialmente el crecimiento general del mercado y están ganando cuota de mercado, reflejando un aumento del 37% en la demanda de servicios en la nube. Los servicios públicos de IaaS y PaaS representaron la mayor parte del mercado y creció un 39% en el primer trimestre (Synergy Research Group, 2020). Una encuesta realizada por IDC indicó que, de 200 líderes de tecnologías de la información, el 27,5% confirmó que la migración a la nube era esencial para el sector empresarial. Sin embargo, el 17% de las empresas ralentizaron sus planes de digitalización debido a la pandemia. Asimismo, el 66% admitió que debido a la pandemia se aceleró la implementación a la nube. De tal manera, que el 85% de líderes admitieron que el cambio permitió a las compañías una mejor posición frente a futuras crisis o problemáticas al proporcionar ahorros en los costos y mayor eficiencia (Frank Della Rosa, 2021).

El instituto de investigación Gartner pronosticó un crecimiento mundial del 18,4% para el año 2021. Asimismo, la infraestructura PaaS tendrá un crecimiento del 26,6% (Gartner, 2020). Los servicios en la nube cada vez están siendo más usados por las empresas y diversos sectores en la industria, permitiendo desarrollar aplicaciones de alto rendimiento y escalables.

Según Satya Nadella, durante la pandemia el cloud computing ha tenido una migración significativa. Asimismo, en el ámbito del desarrollo de software GitHub ha tenido un aumento del 25% de colaboraciones durante los primeros meses de la pandemia (Konrad, 2020). De la misma forma, las soluciones tecnológicas a través del desarrollo de software se han incrementado y tenido mayor importancia en la industria, resolviendo problemas de servicios, conectividad y disponibilidad. No obstante, las empresas encargadas de brindar estos servicios

de software han tenido la necesidad de mejorar su escalabilidad, disponibilidad y tolerancia a fallos en sus sistemas brindados, dado que la demanda sobre este tipo de sistemas se ha visto en aumento. Por consiguiente, la presente investigación pretende desarrollar una metodología de conversión de una aplicación monolítica a microservicios; para reducir la complejidad se realizó un análisis de los posibles métodos que serán tomados para el proceso de desarrollo de la metodología.

1.1. Realidad Problemática.

Las plataformas sobre las que se desarrolla software están en constante mejora y desarrollo, John McCarthy en el año de 1961 sugirió que los avances en la informática nos conducirán a que en algún momento la computación sea tan importante que se contará con una un servicio público, de hecho esto no estuvo lejos de la realidad, a finales de los años 90 Amazon se percató que no utiliza toda su capacidad en cuanto a su infraestructura informática, por lo que en el año 2002 empieza a ofrecer un servicio llamado Amazon Web Services (AWS), este fue el comienzo del gigante tecnológico Amazon, con servicios en Cloud Computing, permitiendo a sus clientes agilidad y disminución de costos (MakeSoft, 2016).

Un estudio realizado por Destino Negocio en colaboración con Movistar nos muestra en una infografía la perspectiva del Cloud Computing detallando los beneficios, tales como la optimización de personal, velocidad en los negocios, eficiencia en procesos, confiabilidad en operaciones y sostenibilidad. Estos beneficios son muy importantes, siendo una de las principales razones para que las pequeñas y medianas empresas (Pymes) migren paulatinamente hacia la nube, a esto se suma las copias de seguridad, ahorro en infraestructura y accesos remotos, a sí mismo el 55% de empresas ven como prioridad la nube para el desarrollo y despliegue de sus aplicaciones y/o servicios (Negocio, 2016).

Las Pymes hacen uso de las Tecnologías de la Información (TI) para poder

obtener potenciar su crecimiento. Además de mejorar sus procesos para poder organizar y tener un mejor control de sus datos, además de reducir sus costos entre un 15% y 20% el tiempo de los procesos operacionales también disminuyen, esto sucede solo si el software que utiliza la empresa cumple con sus necesidades. Asimismo, los datos reflejados por Perú 21 nos muestran que en temas de porcentaje las compañías que hacen uso de un software de gestión son representadas por el 55%. Por otra parte, las empresas que desarrollan software a medida están representadas por el 21%. (Perú21, 2016).

PwC CEO Survey, muestra en un estudio realizado que el 86% de los directores ejecutivos en los próximos 5 años ven a las Tecnologías de la información como prioridad y tendencia. La alianza impulsada por Microsoft y Soluciones Orión se centran en que las empresas peruanas tengan un impulso para lograr una transformación digital. La transformación digital aporta sustancialmente a la empresa en la medida que este se sigue sus lineamientos, por ello es clave para lograr alcanzar los objetivos, pero por otra parte no contar con la evolución tecnología puede suscitar un limitante para su crecimiento y su desarrollo. Por ello, la alianza formada busca iniciar la transformación digital mediante una serie de diez servicios que se orientan a las empresas para potenciar la transformación digital, inversiones en la nube y/o repotenciar sus plataformas. Entre los servicios que se ofrecen se encuentra Azure Microservices – Containers para la adopción a los microservicios y los contenedores, servicios para visión estratégica en la nube sobre las infraestructuras desplegadas a través de Azure con alta disponibilidad y tolerancia a fallos (Microsoft, 2017).

Las empresas que desarrollan software o que lo distribuyen, tienden a ofrecer sus sistemas de manera remota ya sea en un servidor dedicado o en la nube, como anteriormente abordamos el tema de Cloud Computing, las aplicaciones que se alojan ahí cuentan con muchas ventajas, pero además son desarrolladas de diversas maneras, y ofrecidas a los clientes como servicios, estas aplicaciones pueden estar desarrolladas de manera

tradicional, como la interfaz de usuario en el lado del cliente (Front-End), la lógica de negocio y el acceso a los datos por el lado del servidor (Back-End), a esto se le llama una aplicación monolítica.

Una aplicación desarrollada de manera monolítica consta de una capa de usuario y una de acceso a datos combinadas y sobre una plataforma, esto quiere decir que es independiente, además de que engloba toda la lógica para que su función sea la correcta no es modularizada, tiende a tener una estructura simple y que toda la funcionalidad se encuentre en un código único, esto no significa que sea lo adecuado al crecer en escalabilidad, desarrollo, disponibilidad y carga.

Según Richardson, el desarrollo de una aplicación empresarial de comercio electrónico de manera monolíticamente consta de una interfaz de usuario donde se toma en cuenta la User Interface (UI) y User Experience (UX), lógica de negocio las cuales determinan las reglas para de cómo se debe utilizar la información, acceso a datos y la base de datos en la cual se logra la persistencia de los datos, su despliegue será mediante un archivo war. Entre los beneficios se encuentra el simple desarrollo e implementación además de la escalabilidad, pero esta de basa en generar varias copias de toda la aplicación detrás de un balanceador de carga. Sin embargo, a medida que el equipo y la aplicación crece se empieza a notar ciertos inconvenientes, tales como el gran manejo del código base, la implementación continua, la ampliación, la escalabilidad y el compromiso con la tecnología que ha servido para su desarrollo., esto dificulta poder implementar otro tipo de tecnologías para sus posteriores desarrollos e implementaciones (Richardson, 2018b).

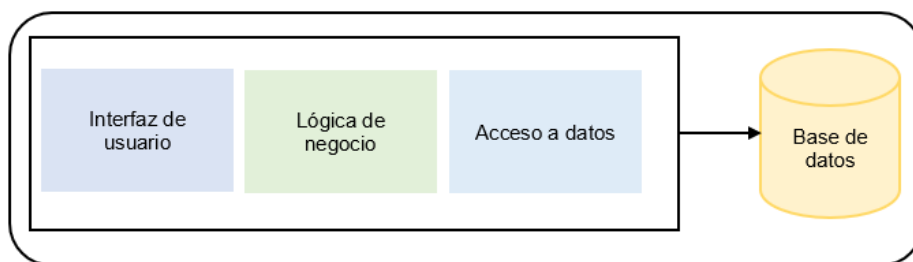


Figura 1. Representación de una aplicación monolítica. Fuente: Elaboración propia.

El desarrollo de una aplicación monolítica tiende a lidiar con la capacidad del servidor donde se está ejecutando, además del exceso de usuarios y falta de modularidad, para tener una idea de la arquitectura podemos observar la Figura 1, para ello la solución es desarrollarla utilizando microservicios, pensemos en la aplicación como bloques de construcción, esto nos permitirá mejor escalabilidad, disponibilidad, versatilidad, mejoras continuas (Richardson, 2018a).

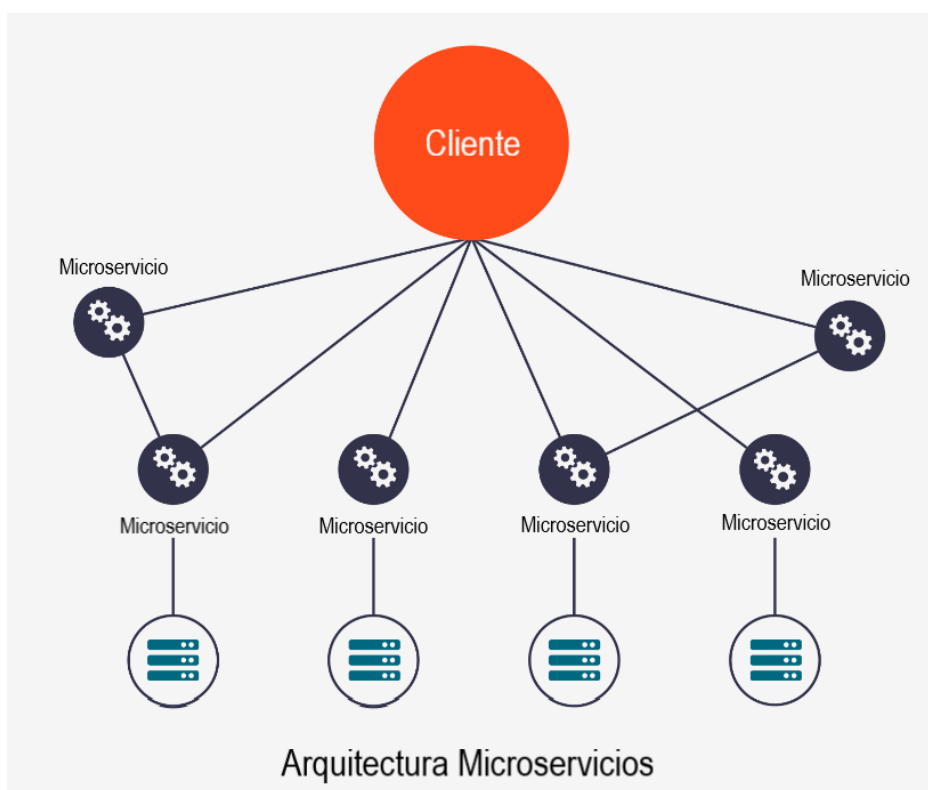


Figura 2. Arquitectura microservicios. Fuente: (Anita, 2016).

(Fan & Ma, 2017) manifiesta como podemos llevar una aplicación móvil monolítica a microservicios, para ello el proceso de conversión se realizó con la extracción de la lógica de negocios mediante Domain Driven Design (DDD), esto con la finalidad de realizar el análisis de la arquitectura, la cual nos llevará a la identificación de los microservicios candidatos, ya que DDD promueve un diseño bajo acoplamiento, de tal forma que cada uno opere de manera independiente.

Llevar una aplicación monolítica a microservicios tiene ventajas, como las que hemos visto previamente, pero también conlleva un proceso de cierta complejidad desde las preguntas, ¿Cómo?, ¿Por qué?, ¿Para qué?, ese es el enfoque en el cual es necesario centrarse, para llevar un proceso adecuado de conversión a microservicios.

1.2. Trabajos previos.

(V. Singh & Peddoju, 2017) en su investigación "*Container-based microservice architecture for cloud applications*", manifiesta que la virtualización de los microservicios son claves para aumentar el rendimiento de las aplicaciones que son desplegadas en la nube, para ello se debe de realizar el despliegue de microservicios en Máquinas virtuales (VM's), sin embargo, esto muestra un consumo de recursos considerable.

Las máquinas virtuales simulan un sistema operativo por lo que supone que fuese una maquina real, la alternativa a las VM's son los contenedores, estos comparten el sistema operativo huésped, además de realizar su virtualización a nivel de núcleo los contenedores se ejecutan como procesos aislados, esto los hace fáciles de administrar, automatizables, escalables, ligeros y con alto rendimiento. Las aplicaciones necesitan ser actualizadas, más aún las empresariales ya que tienen un ciclo de vida muy pequeño, el tiempo de despliegue debe garantizar un mínimo de inactividad de la aplicación.

Se realizó la comparación entre los contenedores y las VM's para el

despliegue de microservicios, se utilizó una aplicación de mensajería y los resultados fueron, que el tamaño de la imagen VM's era de 1056MB, tiempo de despliegue de 10s y el tiempo de actualización de 17s, por el contrario el tamaño de la imagen del contenedor era de 357MB, tiempo de despliegue 2s y tiempo de actualización 5s, En base a los resultados obtenidos se puede comprobar que los contenedores son los lanzadores de despliegue de microservicios más adecuados en comparación de las VM's.

(Buzato, Goldman, & Batista, 2018) en su investigación "*Efficient resources utilization by different microservices deployment models*" manifiesta que en el uso de microservicios se debe de tener en cuenta la cantidad de instancias y contenedores en los que se realiza el despliegue, ya que cuando los contenedores se concentran en una sola instancia da como resultado un excesivo consumo de recursos.

Para realizar el experimento se crearon dos entornos de ejecución en la plataforma de la nube de Google con la siguiente configuración, 1 CPU Intel Xeon, 2.20 GHz con un núcleo, 55 MB de memoria caché, 3.75GB de RAM y disco con 10 GB de almacenamiento, sistema operativo Ubuntu LTS 16.04 y además ejecutando una aplicación de procesamiento de imágenes. Se realizaron dos experimentos uno con un solo servidor y en segundo con tres servidores y cada uno en dos escenarios diferentes, el primer escenario con un solo contenedor y el segundo escenario en dos contenedores separando la capa de aplicación y de datos. Los resultados del primer experimento dieron una considerable reducción de en los recursos de red cuando se utiliza un solo contenedor para microservicios, en los resultados del segundo experimento cuando se utilizó un entorno de tres servidores no se encontraron variaciones evidentes en el uso de recursos como en el experimento anterior.

Concluyendo con los experimentos existe la posibilidad de optimizar recursos mediante un único contenedor donde se encuentre la capa de aplicación y de datos para un microservicio, principalmente la optimización de recursos se da en el uso de la red.

(Waseem & Liang, 2018) en su investigación *“Microservices architecture in DevOps”* manifiesta que hay poco trabajo empírico en arquitectura de microservicios en Development and Operations (DevOps), donde se refleja la poca comprensión de la arquitectura de microservicios, el proceso de migración y el desafío que conlleva llevar una arquitectura tradicional, monolítica a microservicios.

Se Diseñaron y ejecutaron varios estudios de caso en las organizaciones que están involucradas e interesadas en DevOps para el desarrollo de microservicios, las contribuciones que fueron dadas para la solución de este problema fueron, un estudio de mapeo sistemático sobre microservicios en DevOps, un marco basado en la evidencia para sintetizar los desafíos y/o soluciones sobre microservicios en DevOps, recomendaciones para documentar microservicios en DevOps basado en prácticas industriales, el proceso a seguir para realizar una migración de arquitectura monolítica a microservicios se basa en informes empíricos sobre la refactorización de los microservicios en DevOps. En conclusión, realizando la documentación adecuada de microservicios en DevOps se podría obtener una mejor comprensión del amplio panorama que conlleva, desde la migración hasta los desafíos que puede llegar a tener los microservicios.

(Fan & Ma, 2017) en su investigación *“Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report”* manifiesta que microservicios es un sistema de nube emergente que proporciona servicios autocontenidos individualmente, estos son utilizados en la construcción de software's complejos con técnicas de DevOps.

El rápido desarrollo de internet ha aumentado considerablemente la cantidad de peticiones que pueden ser demandadas esto genera complejidad en cuanto al manejo de del software, las aplicaciones tradicionalmente se encuentran conformadas por la interfaz de usuario (cliente), la lógica para el negocio y el acceso a los datos (servidor), estos funcionan como un conjunto en el servidor para servir las solicitudes del cliente, a esta arquitectura se le llama monolítica, esta es muy difícil de gestionar y mantener debido a la falta de modularización, para ello existe la arquitectura de microservicios lo que

permite la construcción por medio de bloques (microservicios). Para realizar una migración de una aplicación que se encuentra con una arquitectura monolítica a microservicios se basan en el SDLC (ciclo de vida de desarrollo de software), esto incluye métodos y herramientas necesarias para el diseño, además de utilizar técnicas de DevOps para automatizar procesos de desarrollo e integración continua. El análisis de la arquitectura tiene un dominio en específico y es que se basan en Domain Driven Design (DDD) para extraer los microservicios candidatos del sistema, además de determinar que el esquema sea consistente en la base de datos y con los microservicios extraídos. Esto indica que en conclusión el artículo plantea un proceso para facilitar la migración de una arquitectura monolítica a microservicios y así poder proporcionar referencias para equipos de desarrollo que busquen realizar la migración.

(Phain & Limpiyakom, 2018) en su investigación *“Scaling Network Traffic Logger with Microservice Architecture”* manifiesta que la arquitectura monolítica sigue siendo una buena opción para muchas aplicaciones, pero tiene una deficiencia la cual es la escalabilidad, por lo que un cambio realizado en una pequeña parte de la aplicación requiere que todo el monolito sea reconstruido y desplegado. En consecuencia, es difícil mantener una buena estructura a lo largo del tiempo.

El modelo de escala en microservicios se llama escala de cubo, dicha escala se representa en tres ejes, el eje X, eje Y, eje Z. El escalado en el eje X se basa en generar copias de una aplicación detrás de un balanceador de carga, este enfoque es simple y utilizado comúnmente para escalar aplicaciones, esto representa una aplicación monolítica, el escalado en el eje Y se basa en ejecutar copias idénticas de la aplicación, esto se debe a que divide la aplicación en muchos servicios diferentes, cada servicio es aislado y responsable de una o más funciones relacionadas,, por lo que se podría decir que la arquitectura en microservicios es un escalamiento en el eje Y. El escalado en el eje Z se basa en ejecutar una copia idéntica de la aplicación, en esto es similar al eje X, pero la diferencia es que cada servidor es responsable de un subconjunto de datos, esto se debe a que cada

componente del sistema es responsable de enrutar cada solicitud con el servidor apropiado, asimismo esta división en el eje Z se usa para el escalamiento de las bases de datos. En conclusiones enfoque general de microservicios recae en eje Y, ya que la escala en Y divide la aplicación en sus componentes y servicios.

(Kargar & Hanifzade, 2018) en su investigación "*Automation of regression test in microservice architecture*" manifiesta que el enfoque de la arquitectura a microservicios es muy beneficioso ya que cada microservicio es independiente, actualizable, escalable y permite una entrega continua, para asegurar una entrega continua es necesario realizar un Test de Software, esto es uno de los desafíos de la implementación automatizada de pruebas de las nuevas versiones que salen al entorno de producción, para ello la Prueba de Regresión es una de las más utilizadas para garantizar la confiabilidad ya que compara dos versiones del sistema basadas en métricas, este metodo permite comparar con la versión anterior en función a métricas medibles como las fallas de sistema, rendimiento, uso recursos, etc. Para realizar la prueba de software se utilizará Jenkins ya que es una herramienta basada en complementos que se implementa utilizando Java. Hay muchas formas para crear estos pasos, como por ejemplo se podría generar un script para la entrega continua. Primero, se preparan los arreglos necesarios para iniciar la operación, creando una cuenta de usuario y definiendo un token en cada proyecto (microservicio) para conectar Git a Jenkins, y luego, para cada microservicio, se crea un elemento en Jenkins. Jenkins es informado de que una nueva versión del sistema está lista para ser construido, cuando los códigos de la rama maestra se envían al servidor git, usando webhook. Esto es cuando comienzan los pasos de implementación continua.

Se utiliza la última versión del código de la rama máster, se verifica la versión de java que se utilizara como base, luego los códigos compilados en Jenkins están disponibles por lo que se deben eliminar los códigos anteriores, la fase de despliegue se da utilizando Maven ya que se crea una version junto con la fecha y hora ya que si se tiene que volver a una versión anterior se puede

realizar, se debe de construir una imagen docker para el microservicio, por lo que la imagen es desplegada en un repositorio, para automatizar la implementación y escalabilidad en entornos de producción se utiliza Kubernetes, la prueba de regresión se define en la entrega continua, para ello se utiliza Diffy una herramienta desarrollada por Twitter a fines del año 2016. Esta herramienta le permite usar el tráfico de entrada como entrada para probar el microservicio desarrollado, lo que podría resultar en una reducción considerable de los costos de las pruebas. Esta herramienta ejecuta la prueba de regresión comparando microservicios. En conclusión, este sistema es basado en la arquitectura de microservicio, y consiste en varios microservicios, algunos de estos microservicios necesitan intercambiar datos para algunas operaciones. Este método automatizado para tomar las Pruebas de Regresión podría utilizarse para reducir los costos de pruebas.

(Djogic, Ribic, & Donko, 2018) en su investigación *“Monolithic to Microservices redesign of event driven integration platform”* manifiesta el rediseño arquitectónico de la plataforma de integración Arquitectura Orientada a Servicios (SOA) siguiendo los principios de microservicio, por lo que debe proporcionar la capacidad de escalabilidad, mejor administración de recursos, mantenimiento y despliegue, para ello es necesario transformar la plataforma de integración para ser microservicios.

La plataforma rediseñada en forma de grupos de microservicios controla las acciones de la base de datos como obtener, agregar, actualizar, eliminar y garantizar la consistencia de las transacciones, la reversión y la actividad de limpieza. Este microservicio se puede utilizar a través de la interfaz de la API REST para garantizar una forma más fácil de consumir DB, los microservicios son lógicamente independientes y cada microservicio contiene su propia configuración para garantizar la capacidad de controlar el paso del flujo de mensajes y algunas otras configuraciones relacionadas con el procesamiento del mensaje. Estos microservicios se basan en la cola, lo que significa que cada servicio tiene su propia cola de alimentación. Este grupo consume la base de datos de transacciones a través del protocolo

HTTP mediante Rest Full.

En conclusión, el rediseño se basa en microservicios y se describe los pasos de la transformación. El rediseño basado en microservicios se elige porque está resolviendo completamente las limitaciones de la plataforma, todos estos beneficios de rediseño de la plataforma que se describen dan más agilidad a los equipos de mantenimiento de la plataforma y son valores clave por los que se selecciona este enfoque en comparación con los otros enfoques para el rediseño de la plataforma.

(Liu, Jiang, Zhao, & Jin, 2018) en su investigación "*A shortest-response-time assured microservices selection framework*" manifiesta que el tiempo de respuesta del servicio se considera como uno de los referentes en cuestión de calidad más importantes de un servicio web. Esta investigación se centra en el problema de selección la ruta de invocación de servicio con el menor tiempo de respuesta. Se propone un nuevo marco de selección de microservicios distribuidos (DMSF) que se puede integrar fácilmente en arquitecturas de microservicios.

Las relaciones de dependencia del servicio se basan en la relación entre el proveedor de servicios y el servicio apuntando en la dirección del servicio de invocación. Microservicios es un nuevo enfoque de diseño inspirado en el servicio, en la que un servicio complejo se compone de uno o más microservicios. Una instancia de microservicio acepta una solicitud de invocación, luego para cada tipo de servicio, se busca en la tabla y selecciona un destino e instancia de servicio que tiene el menor tiempo de respuesta y para cada objetivo instancia de servicio, emite una invocación del servicio. Para evaluar el desempeño de nuestro enfoque, DMSF se implementó una sola máquina virtual VMware (2 núcleos, 2,40 GHz, memoria 2G) con ancho de banda de la red es de 1Gbps. El algoritmo de selección de servicio predeterminado es Enrutamiento distribuido con información de estado local (DLR) y uso del algoritmo Round-Robin para seleccionar el servicio. Para comparación, también se implementó otros dos algoritmos comunes de selección de servicios, tales como, enrutamiento de origen con

información de estado global (SGD) y utilizando el algoritmo de Dijkstra para seleccionar el servicio, además de enrutamiento distribuido con información de estado global (DGD) y usando el algoritmo Dijkstra para seleccionar el servicio. En conclusión, los resultados muestran que DMSF tiene poca complejidad de tiempo y la sobrecarga de comunicación en comparación con SGD y DGD a la vez que garantiza el menor tiempo de respuesta en comparación con DLR.

(Wang & Fahmi, 2018) en su investigación “*Constructing a Service Software with Microservices*” manifiesta que la arquitectura de microservicio (MSA) es una variación de un Arquitectura orientada a servicios (SOA) que permite que un servicio se divida en varios más pequeños, pero de forma simultánea unidades de funcionamiento para que tanto el rendimiento y mantenibilidad de la aplicación puede conseguir una gran mejora, por lo que la concurrencia del sistema, el rendimiento y la reutilización de los microservicios se pueden mejorar utilizando microservicios distribuidos utilizando especificaciones orientadas a objetos (MOOS). Se describe cada subsistema basado en especificación de análisis de objetos, se mapea los subsistemas el subsistema a las plataformas tanto el hardware como el software para saber cómo este reacciona a la concurrencia, además de identificar los recursos globales y se crea su acceso a su mecanismo de control correspondiente, por lo que se describe el control de concurrencia de cada subsistema con la finalidad de que permita a cada subsistema servir múltiples peticiones al mismo tiempo creando un nuevo hilo para cada solicitud aceptada, se definir y reutilizar microservicios basados en subsistemas, se compruebe y actualiza el diseño del sistema verificando si todos los requisitos y problemas de diseño abordados contienen alguna contradicción. Si se encuentra algún error, se regresa al paso correspondiente, se realiza las actualizaciones necesarias y se continúa. En comparación con los enfoques de construcción existentes, MOOS podría aumentar la concurrencia del sistema de aplicación y, por lo tanto, mejorar el rendimiento y la reutilización.

(Guamán, Yaguachi, Cueva C, Jaramillo H, & Soto, 2018) en su investigación *“Performance evaluation in the migration process from a monolithic application to microservices”* manifiesta que la arquitectura de microservicios permite realizar la descomposición del sistema, por lo que el problema se basa en como migrar un sistema monolítico a uno a microservicios. Para llevar un sistema a microservicios se identifica la funcionalidad que cumple el monolito, permite preparar la migración, se identifica las funcionalidades que posee y en base a ellas se construyen los microservicios, además se consigue la comunicación mediante API REST por lo que se pueden utilizar patrones de diseño como Singleton, Proxy. Los servicios se despliegan en su propio contenedor independiente y aislado comunicados por API's. El servidor de aplicaciones aumenta en carga cuando utiliza patrones de diseño, pero es compensado con menos tiempo de respuesta. Las metodologías de desarrollo se deben de ajustar a las necesidades del negocio, esto implica que la integración, entrega y despliegue se debe de ajustar al proceso.

1.3. Teorías relacionadas al tema.

Software

El software permite facilitar las operaciones y/o procesos con la finalidad de resolver las necesidades del negocio, una de sus definiciones según el autor (Pressman, 2010) manifiesta el software como programa aislado que procesa sus funcionalidades controladamente en tiempo real.

Según (Amo, Normand, & Pérez, 2005) manifiestan que el software es representado por un conjunto de programas desarrollados a través de un entorno computacional, estos programas se suelen clasificar en tipos tales como, los programas de control el cual sirve para controlar y supervisar las tareas y procesos, los programas de procesos los cuales sirven para crear programas propios y los de aplicación que son desarrollados por el usuario para resolver problemas específicos.

Ingeniería de Software

Es una disciplina que se basa en la producción del mismo, desde la etapa de planificación, desarrollo, hasta mantenimiento, esto en busca de tener unos óptimos resultados de calidad, costes y confiabilidad en el tiempo establecido, según (Sommerville, 2011) manifiesta que para lograr producir un software de alta calidad se debe de optar por un enfoque organizado y sistemático. Los procesos de software ayudan a obtener los resultados esperados, esto se logra teniendo en cuenta ciertas actividades como la especificación, diseño, implementación, validación y evolución del software. Como ejemplos de estos procesos de puedeos considerar a Rational Unified Process (RUP) que ha sido el más utilizado.

Procesos de desarrollo de Software

Los procesos de desarrollo de software permiten realizar un análisis correcto sobre los requerimientos, seguido por el diseño, construcción y prueba de un sistema se debe de conocer el proceso correcto para poder realizar un buen desarrollo del software, este proceso se debe o no orientar a el desarrollo ágil de software. Los requerimientos cambiantes permiten el descubrimiento de nuevos problemas, en el requerimiento mismo, en el diseño o incluso en la implementación, esto genera inconvenientes a tal magnitud que el sistema tiende a reelaborarse y realizar las pruebas pertinentes nuevamente. Existen ciertos procesos de desarrollo de software que prolongan la entrega final al cliente, estos procesos como el Desarrollo en Cascada o RUP son muy rigurosos en sus etapas ya que para iniciar una nueva etapa se debe de esperar la finalización de la anterior, estos procesos no tienden a una estructura cambiante en el concepto del negocio. Sin embargo, un enfoque ágil de desarrollo de software se basa en la tendencia al cambio que se puede llegar a generar centrandó en la flexibilidad que puede aportar bajo ciertas circunstancias con una capacidad de respuesta que se adapte al proyecto y su desarrollo. (Sommerville, 2011).

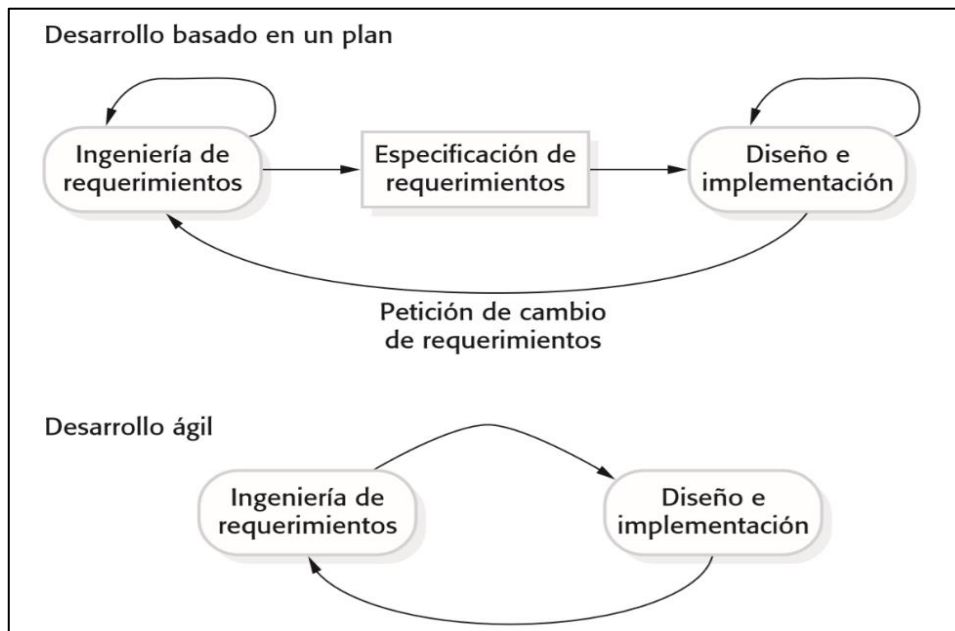


Figura 3. Desarrollo basado en un plan y desarrollo ágil. Fuente: (Sommerville, 2011).

Metodología Ágil

Historias de Usuario

Las metodologías ágiles de desarrollo de software tienen una percepción de cambios inherentes, esto con respuesta a las necesidades que son generadas por el cliente. Para ello se debe de simplificar las tareas de gestión de las mismas y tener una documentación de las necesidades del sistema, esto con el fin de involucrar al cliente durante el proceso de desarrollo para lograr una comunicación frecuente y directa con un feedback constante para mejorar las planificación y especificaciones del requisito del sistema (Cao & Ramesh, 2008).

En las historias de usuario se deben de especificar las necesidades del sistema, esto servirá como guía durante el desarrollo del software, ya que para ello se debe detallar lo requerido en un lenguaje que el cliente y desarrollador puedan comprender, dichos requisitos se dividen en tareas que se deberán de realizar para cumplir con las funcionalidades solicitadas por el cliente, además que se

debe de realizar una estimación de tiempo en la cual se acuerde como aproximación la duración del desarrollo de la funcionalidad (Cao & Ramesh, 2008).

Programación en parejas

Es una práctica que ayuda a acelerar el proceso de desarrollo ya que permite tener constante comunicación y dividir lo planificado, verificando mejoras, errores o alternativas que logren aumentar la productividad y disminuir el tiempo en el que se desarrolla las funcionalidades. Sin embargo, esto puede convertirse en un gran inconveniente cuando las habilidades de la pareja difieren entre sí, esto puede llegar a ser frustrante para ambos desarrolladores por lo que en vez de ser una ventaja llega a ser una desventaja para el desarrollo en sí. Por lo que la programación en parejas debe de comprender como una técnica de aprendizaje y crecimiento mutuo (Baumeister, Marchesi, & Holcombe, 2005).

Integración continua

La integración continua permite integrar cada cambio que se logra introducir durante el desarrollo, para que esto se realice de la mejor manera se sugiere disponer de test de tal manera que en nuevo código introducido este integrado con el resto del sistema sin ocasionar algún fallo inesperado a consecuencia del nuevo módulo, asimismo se pueden desarrollar test automatizados para realizar las pruebas, de esta manera se lograría integrar de manera eficiente muchas veces en un mismo día, mejorando ciertas circunstancias como la complejidad durante la codificación e integración (Hasselbring & Steinacker, 2017).

Refactorización

La refactorización permite mejorar el sistema sin llegar a influir en el código funcional actual, permite reestructurar el código para mejorar en gran medida, haciéndolo más flexible, simple, legible para los demás desarrolladores, entre

otras características. El código es parte de las metodologías ágiles por lo que se necesita que permita legibilidad y comunicación entre equipos o desarrolladores (Nakazawa, Ueda, Enoki, & Horii, 2018).

Scrum

Scrum para el desarrollo de software fue presentada en una conferencia de la mano del presentador Ken Schwaber, el cual presentó a la metodología tomando como base el concepto de Scrum. Los principios de desarrollo ágil son tomados por Scrum a través de un conjunto de prácticas y reglas, las cuales ayudan a la evolución del producto lográndose adaptar a las necesidades, teniendo en cuenta la calidad del resultado basándose en las estrategias de desarrollo a través de iteraciones (sprint's), según (Menzinsky, López, & Palacio, 2019) manifiesta que a partir de las funcionalidades y los detalles que se desean obtener se debe de generar un producto backlog con historias de usuario, esto deriva en una planificación del sprint y por lo consiguiente en un sprint backlog, por lo que al finalizar cada sprint se realiza una entrega del producto operativo, esto se realiza incrementalmente, cada sprint puede tener una duración máxima de hasta 6 semanas, por lo que nos sugiere Scrum es que los sprint's no excedan el tiempo de 4 semanas (1 mes). Diariamente se debe de realizar una monitorización de cómo se va desarrollando el sprint, esto se logra mediante reuniones donde se realiza una revisión sobre el trabajo realizado el día anterior y el trabajo que se realizará, estas reuniones deben tener lugar en tiempos muy cortos, aproximadamente 5 a 15 min como máximo, a estas reuniones se les llama Scrum diario (daily scrum).

Siguiendo la estructura de SCRUM podemos definir que su base son las historias de usuario, esto permite describir las características esperadas del software en desarrollo, por lo que en las mismas se pueden incorporar cuestiones relacionadas a rendimiento, restricciones, funcionalidades, entre otras. Dichas historias representan la visión del usuario por lo que se debe de utilizar un lenguaje común para lograr la completa comprensión de lo que se busca, primero de busca determinar quien propone la historia de usuario, luego las características y la razón por la que es necesaria, esto se realiza se manera

conjunta con el cliente, donde se defina de manera clara y concisa que es lo que se requiere, esto permite servir como guía para el desarrollo. Las historias de usuario no se ven obligadas a describir de manera detallada lo que se necesita, esto permite que se puedan hacer ciertas modificaciones dependiendo de las necesidades del usuario. El resultado de esto se denomina Product Backlog, el cual contiene todas las tareas e historias de usuario con su prioridad respectiva para el desarrollo. Durante el Product Backlog se pueden aclarar dudas respecto a las historias de usuario que no se resolvieron satisfactoriamente, por lo que se pueden aclarar en el instante. Consecuentemente se realizan los sprints los cuales deben tener una corta duración de aproximadamente 30 días como máximo. Para cada iteración se realiza una reunión donde se planifica lo que se llevará a cabo y su fecha de finalización respectiva, el conjunto de tareas a desarrollar durante el sprint se le denomina Sprint Backlog, el cual mueve las tareas e historias de usuario de mayor prioridad para su realización.

De manera iterativa diariamente se debe de realizar una reunión de no mas de 15 minutos donde se realice cierta revisión de lo que se hizo ayer, que se hará hoy y que se necesita para lograrlo, esto con el fin de identificar los obstáculos que impiden el ciclo de avance normal de desarrollo. Ya finalizado el Sprint se debe de obtener un entregable para mostrar lo desarrollado durante el sprint, esto procede a una revisión para saber si se cumple con los requerimientos solicitados y si en todo caso no se cumpla se pueda hacer una nueva historia de usuario, el entregable debe de cumplir al 100% por lo requerido en la historia de usuario y haber superado los test. Al finalizar el sprint se debe de realizar una reunión, la cual se centra en una retrospectiva para lograr mejorar las debilidades o inconvenientes en el próximo sprint.

Para lograr un proceso adecuado de SCRUM se deben de tener en cuenta diferentes papeles, los cuales están conformados por el Product Owner el cual es el encargado de la dirección de las historias de usuario y que estas se cumplan totalmente, el Scrum Master es el responsable del desarrollo del proyecto el cual tiene las directrices para dirigir el flujo de SCRUM, el Scrum

Team que es conformado por las personas encargadas de implementar las funcionalidades definidas, con la finalidad de cumplir el objetivo del sprint. En teoría se estima que Scrum debería estar conformado con 2 personas como mínimo y como máximo 7 u 8 personas (Menzinsky et al., 2019).

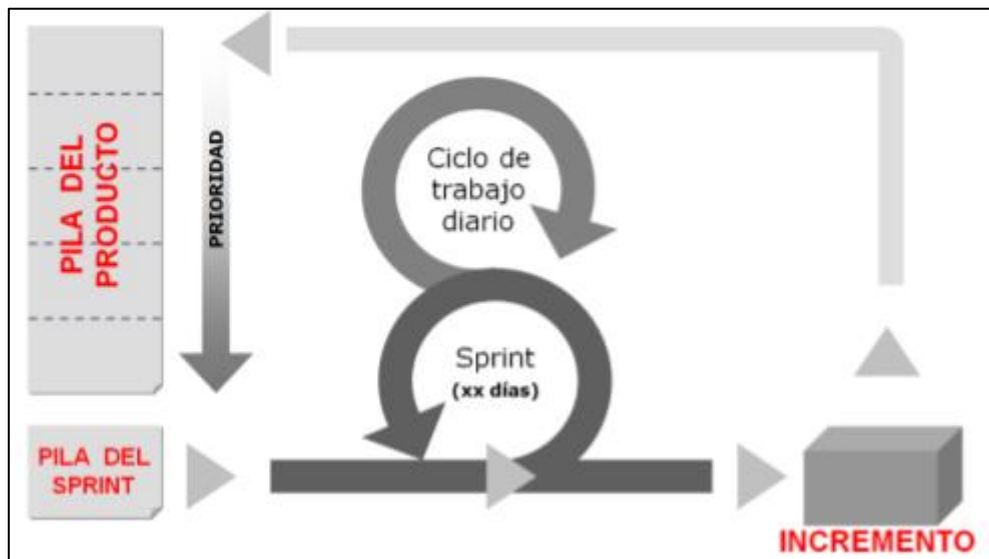


Figura 4. Diagrama del ciclo interactivo Scrum. Fuente: (Menzinsky et al., 2019).

Extreme Programming

Extreme Programming (XP) se define como una disciplina la cual se enfoca en lograr objetivos en equipo, estos equipos producen un software de calidad a un ritmo continuo y sostenible, siguiendo con los objetivos de XP se debe llevar responsabilidad y transparencia al desarrollo de software, además de lograr un desarrollo eficiente y con mucho menos defectos. Esto se logra con una filosofía de desarrollo, valores, comunicación, retroalimentación, simplicidad y respeto. XP se distingue de otras metodologías por sus cortos ciclos de desarrollo que ofrecen una retroalimentación continua, su planificación y evolución a lo largo de la vida del proyecto, demostrando así que se basa en buenas prácticas. En la [Figura 6](#) podemos observar la representación de la metodología XP (Beck & Andres, 2004).

Las metodologías ayudan a desarrollar software de diferentes maneras, monóticamente, distribuida, microservicios, por lo tanto, debemos de analizar que metodología utilizar y en los casos se pueden aplicar correctamente.

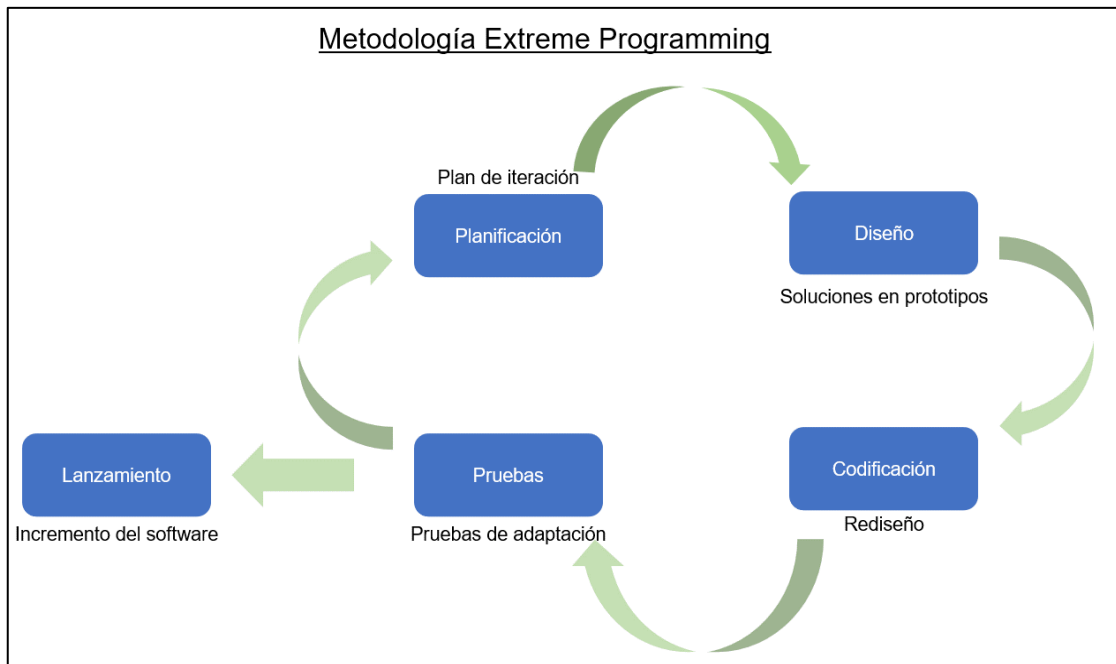


Figura 5. Metodología Extreme Programming. Fuente: Elaboración propia.

Arquitectura Monolítica

La arquitectura monolítica representada por una aplicación se desarrolla de manera de bundle, esto se debe a que consta de una interfaz de usuario representada por el Front-End, lógica de negocio representadas por el Back-End y la base de datos, todo sobre un código único, de esta manera el despliegue se hace mediante un solo archivo. Las ventajas que proporciona desarrollar software de manera monolítica son, el simple desarrollo, implementación y la escalabilidad, pero esta genera varias copias de toda la aplicación en distintos servidores y mediante un balanceador de carga, lo que es un conflicto a largo plazo, por el contrario, sus desventajas radican en que

requiere un compromiso con la pila tecnológica en la que ha sido desarrollada, la implementación continua deficiente, ampliación de la aplicación difícil y obstáculo para el desarrollo escalado (Richardson, 2018b).

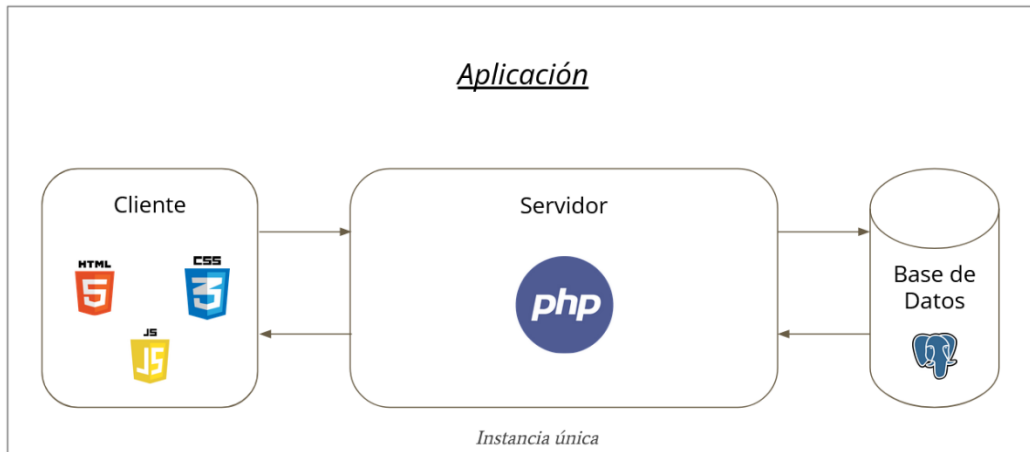


Figura 6. Representación de una aplicación monolítica en una instancia.
Fuente: Elaboración propia.

En general, el código de una aplicación monolítica grande es el más difícil de entender para los desarrolladores, y lleva tiempo antes de que un nuevo desarrollador pueda ser productivo. Además de que cargar una aplicación monolítica es pesado para el IDE esto hace que el desarrollador sea menos productivo, otra de los inconvenientes de esta arquitectura es que aplicar un cambio es difícil de implementar y lleva mucho más tiempo debido a la gran base de código que se tiene, lo cual ocasiona que haya un alto riesgo de errores si es que antes no se analiza de manera adecuada el impacto que se puede generar. Por lo tanto, se convierte en un requisito previo para que los desarrolladores realicen un análisis de impacto exhaustivo antes de implementar cambios. En aplicaciones monolíticas, las dependencias se acumulan con el tiempo a medida que todos los componentes se agrupan. Por lo tanto, el riesgo asociado con el cambio de código aumenta exponencialmente a medida que aumenta el código, lo cual genera que cuando una base de código sea enorme y muchos desarrolladores están trabajando en ella, se vuelve muy difícil construir productos e implementar nuevas características. Para ello se debe asegurar de que todo esté en su lugar y de que todo esté coordinado. Netflix, el proveedor de transmisión de Internet, tuvo problemas para desarrollar

su aplicación con alrededor de 100 personas. Luego, usaron una nube y dividieron la aplicación en partes separadas. Estos terminaron siendo microservicios. Los microservicios surgieron del deseo de velocidad y agilidad y de desplegar equipos de forma independiente (Gonzalez, RV, & Sharma, 2017).

Escala de Cubo

Las aplicaciones tienden a escalar de tal forma que en algunas ocasiones esto puede suponer cierta desventaja, sin embargo, para ello existen tres tipos de escalas, en el eje X, Y, Z. El escalado en el eje X se basa en generar copias de una aplicación detrás de un balanceador de carga, este enfoque es simple y utilizado comúnmente para escalar aplicaciones, esto representa una aplicación monolítica, por lo que los inconvenientes de este enfoque se dan debido a que cada copia accede a todos los datos, la cache, en consecuencia, requieren más memoria para ser efectivos, además este enfoque no aborda los problemas de aumentar la complejidad en el desarrollo. El escalado en el eje Y se basa en ejecutar copias idénticas de la aplicación, esto se debe a que divide la aplicación en muchos servicios diferentes, cada servicio es aislado y responsable de una o más funciones relacionadas, para descomponer la aplicación en servicios se utiliza la descomposición de verbos que solo implementa un caso, por ejemplo pagos, la otra opción es descomponer por nombre y así crear servicios responsables de las operaciones de una entidad, también se podría utilizar la combinación de a descomposición basado en el verbo y el sustantivo, por lo que se podría decir que la arquitectura en microservicios es un escalamiento en el eje Y. El escalado en el eje Z se basa en ejecutar una copia idéntica de la aplicación, en esto es similar al eje X, pero la diferencia es que cada servidor es responsable de un subconjunto de datos, esto se debe a que cada componente del sistema es responsable de enrutar cada solicitud con el servidor apropiado, asimismo esta división en el eje Z se usa para el escalamiento de las bases de datos, por ejemplo, si tengo la clave primaria de la tabla Carro esta se puede usar para dividir las filas entre dos servidores de bases de datos diferentes, por lo que podemos utilizar replicas/esclavo (Abbott & Fisher, 2015).

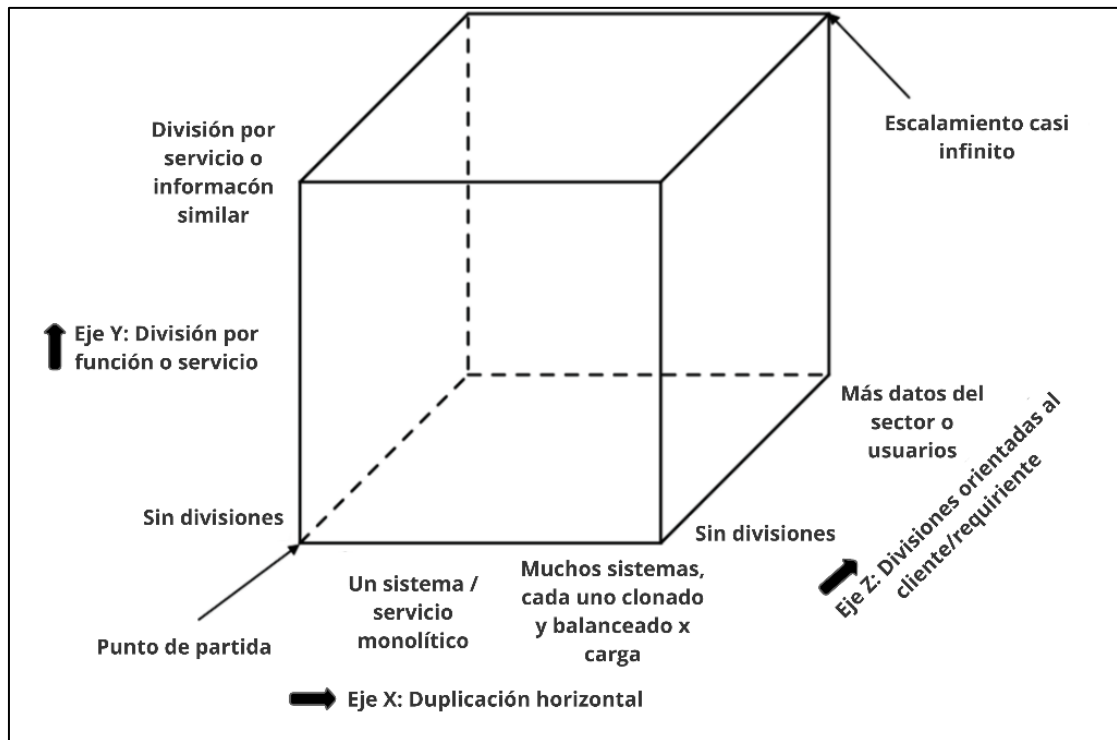


Figura 7. Escala de cubo. Fuente: (Abbott & Fisher, 2015).

Arquitectura Microservicios

Según (Richardson, 2018a) manifiesta que las aplicaciones monolíticas lidian con el servidor en el que se está ejecutando, para ello existe la arquitectura basada en microservicios, pensemos en la aplicación monolítica, pero ahora como si estuvieran formadas por piezas de construcción, esto nos permitirá versatilidad, escalabilidad y mejoras continuas (Vernon, 2016).

Los microservicios han ganado mucha tracción recientemente. Una arquitectura basada en microservicios es una forma de diseñar su software. En dicha arquitectura, las aplicaciones se dividen en servicios más pequeños para que se puedan implementar y administrar por separado. Esto elimina muchos puntos débiles que ocurren en las aplicaciones monolíticas tradicionales. Dicho esto, los microservicios se pueden construir utilizando cualquier lenguaje de programación. De hecho, hay muchas bibliotecas y marcos que ayudan a los programadores a crear microservicios utilizando Java, Scala, C #, JavaScript, Python, Ruby, entre otros lenguajes (Murugesan, 2017).

La propuesta de microservicios para mejorar la capacidad de software es muy atractiva. Esto se debe a que no es fácil que un software sea susceptible al cambio, para ello microservicios está centrado en la construcción de componentes reemplazables, pero cuando se quiere adoptar a arquitectura a microservicios , esto se debe a que surgen ciertas incertidumbres y sobre si este enfoque puede causar un daño potencial a los sistemas, por lo que se identifican con frecuencia ciertos problemas, tales como, la gestión y la coordinación, esta sería demasiado difícil, además de que el estilo de los microservicios no tienen en cuenta su contexto, entorno y requisitos único (Nadareishvili, Mitra, McLarty, & Amundsen, 2016).

Según (Newman, 2015) los microservicios son pequeños servicios autónomos que trabajan juntos, además, según Adrian Cockcroft reafirma este pensamiento manifestando que la arquitectura basada en microservicios es de acoplamiento libre con contextos limitados. Por ello gestionar esta arquitectura es más complejo, pero para resolver la complejidad se pueden utilizar herramientas y automatización de procesos, esto ayudará al rendimiento, ya que es beneficioso para la empresa. Para tener una conceptualización del modelo de diseño de microservicios nos podemos ayudar de la Figura 8 donde se muestran cinco partes, las cuales consta del servicio, solución, proceso, herramientas, organización y cultura (Nadareishvili et al., 2016).

Los microservicios se desarrollan gracias a las exposiciones de API, estas se pueden probar continuamente para su integración continua. Ya que los cambios son pequeños y adecuados a las necesidades en cada ciclo de lanzamiento continuo se puede hacer uso de pruebas de regresión, de esta manera se logra corregir los defectos que se encuentran eventualmente, esto logra reducir el riesgo de una mala implementación. Esto da como resultado una mayor velocidad con un menor riesgo asociado. Debido a la separación de la funcionalidad y el principio de responsabilidad única, los microservicios hacen que los equipos sean muy productivos y tengan una mejor motivación y claridad en la implementación de funciones, esto permite un ciclo de retroalimentación

más corto y una priorización mejor y rápida de la canalización de características. Un ciclo de retroalimentación más corto hace que la detección de defectos también sea más rápida. Cada equipo de microservicios trabaja de forma independiente y se pueden implementar nuevas características o ideas sin coordinarse con un público más amplio. La implementación del manejo de fallas de punto final también se logra fácilmente en el diseño de microservicios. (Gonzalez et al., 2017)



Figura 8. Modelo de diseño de microservicios. Fuente: (Nadareishvili et al., 2016).

Domain-Driven Design (DDD)

El enfoque DDD permite utilizar una estructura de prácticas que ayudan en enfocar y acelerar desarrollos complejos para ayudar en la toma de decisiones. Consiste abordar un dominio central, tener un idioma omnipresente dentro del contexto, los beneficios son notables al tener convenciones de modelado rigurosas. DDD se basa en el dominio, el cual se representa mediante una esfera de conocimiento, influencia o actividad. El área temática a la que el usuario aplica un programa es el dominio del software. (Evans, 2014).

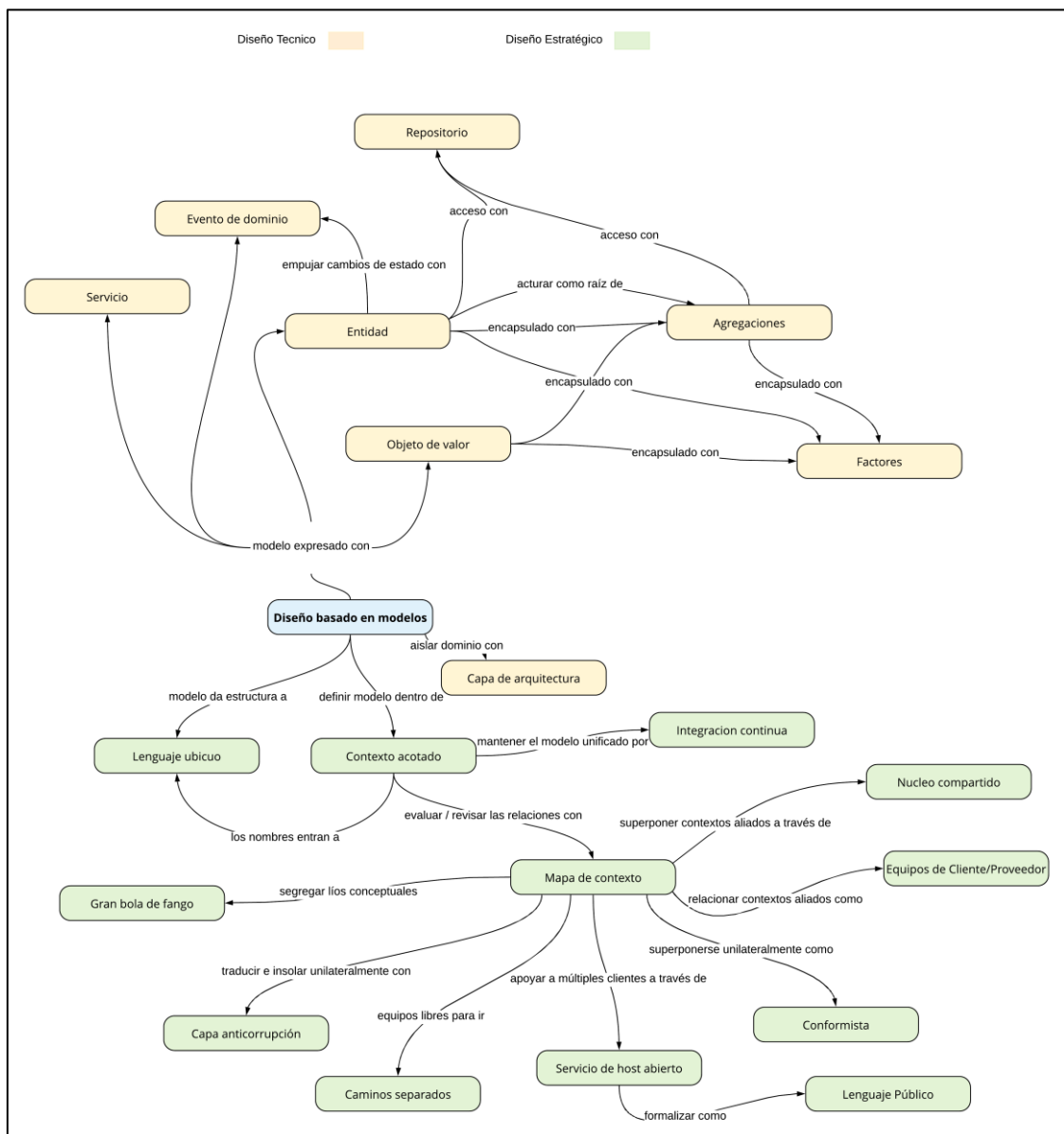


Figura 9. Descripción general del lenguaje de patrones. Fuente: (Evans, 2014).

El modelo de dominio comprende 2 diseños el técnico y el estratégico, de los cuales se conceptualizarán los siguientes:

Diseño Técnico

Entidad

Los objetos representan una continuidad e identidad, pasando por un ciclo de vida donde los atributos pueden cambiar, asimismo Un objeto debe distinguirse de otros objetos, aunque puedan tener los mismos atributos. La identidad equivocada puede conducir a la corrupción de datos. Por lo tanto, un objeto se debe de distinguir por su identidad, en lugar de sus atributos, se debe de mantener la definición de clase simple y centrada en la continuidad y ciclo de vida (Evans, 2014).

Objeto de valor

Si pensamos en esta categoría de objeto como solo la ausencia de identidad, no hemos agregado mucho a nuestro vocabulario. De hecho, estos objetos tienen características propias y su propia importancia para el modelo. Estos son los objetos que describen cosas, cuando solo interesan los atributos y la lógica de un elemento del modelo se debe de clasificar como un objeto de valor, por lo que debe de expresar significado de los atributos que transmite y dale funcionalidad relacionada, se debe de tratar al objeto de valor como inmutable, las operaciones se deben realizar sin efectos secundarios que no dependan de ningún estado mutable. No se le debe dar identidad a un objeto de valor, con el fin de evitar las complejidades de diseño necesarias para mantener las entidades (Evans, 2014).

Evento de Dominio

Se necesitan, Pero si necesita conocer las causas reales de los cambios de estado que se generan, esto generalmente no es explícito, y puede ser difícil explica. Las pistas de auditoría pueden permitir el rastreo, por lo que un historial de cambios de las entidades puede permitir conocer los estados anteriores. En los sistemas distribuidos surge un conjunto distinto de problemas relacionados.

El estado de un sistema distribuido no puede mantenerse completamente consistente en todo momento, por ello es necesario propagar los cambios que se han realizado. Por lo que se debe de saber las actividades del dominio como una serie de eventos discretos. Representar cada evento como un objeto de dominio. Estos eventos del sistema que reflejan la actividad dentro del propio software sirven como una forma de llevar información sobre el evento de dominio al sistema, un evento de dominio es una parte completa del modelo de dominio, representando algo que sucedió en el dominio (Evans, 2014).

Diseño Estratégico

Lenguaje Ubicuo

Es un lenguaje estructurado que gira alrededor del modelo de dominio, dicho lenguaje debe ser usado por los miembros del equipo en el mismo contexto, con la finalidad de lograr una conexión no solo limitándose a el uso con el cliente si no también en la comunicación de los equipos de desarrollo como lo puede ser FrontEnd, BackEnd, entre otros, logrando el entendimiento común entre todos (Evans, 2014).

Contexto

Equipos que trabajan independiente pueden resolver el mismo problema en diferentes maneras por falta de comunicación. A menudo no está claro en que contexto se debe de aplicar un modelo, por lo que se debe definir explícitamente el contexto dentro del cual se aplica un modelo, estableciendo límites en términos de organización del equipo, uso dentro de partes específicas de la aplicación y manifestaciones físicas como bases de código y esquemas de bases de datos. Se debe de aplicar Integración continua para mantener los conceptos y términos del modelo estrictamente consistentes dentro de estos límites, estandarizando un solo proceso de desarrollo dentro del contexto, que no necesita ser usado en otro lugar (Evans, 2014).

Integración continua

Consiste en instituir un proceso de fusión de todo el código y otros artefactos de implementación con frecuencia, se puede realizar con pruebas automatizadas. Se debe de ejercer el lenguaje ubicuo para tener una visión compartida del modelo a medida que los conceptos evolucionan en la percepción de diferentes personas (Evans, 2014).

Publicación del lenguaje

La traducción entre los modelos de dos contextos limitados requiere un lenguaje común. Probablemente los modelos están indocumentados. Por lo que se recomienda hacer uso de un lenguaje compartido bien documentado que pueda expresar la información de dominio necesaria como un medio común de comunicación, traducándose según sea necesario en y fuera de ese idioma. Asimismo, el idioma establecido puede ser publicado en forma de estándares de intercambio de datos. Los equipos de proyecto también desarrollan los suyos para su uso dentro de su organización. El lenguaje publicado a menudo se combina con el servicio de host abierto (Evans, 2014).

El uso de DDD evita el diseño de aplicaciones monolíticas, para ello se utilizan al menos dos herramientas fundamentales para el diseño estratégico. Una de ellas es el contexto limitado y el lenguaje ubicuo. En el caso del contexto limitado se debe de responder la pregunta “¿Qué es el núcleo?”, este contexto debe contener todos los conceptos fundamentales para la iniciativa estratégica y expulsar las demás. Esto significa que algunos conceptos estarán en contexto y se incluirán claramente en el idioma del equipo, ya que son parte del lenguaje ubicuo del equipo propietario del contexto acotado. Los expertos en dominios, naturalmente, estarán más centrados en las preocupaciones comerciales. Sus pensamientos se centrarán en su visión de cómo funciona el negocio. En que el dominio de Scrum, cuente con que el experto en dominios sea un Scrum Master o similar que comprenda a fondo cómo se ejecuta Scrum en un proyecto.

Por otro lado, los desarrolladores se centran en el desarrollo de software. Como se muestra aquí, los desarrolladores pueden consumirse por los lenguajes y tecnologías de programación. Sin embargo, los desarrolladores que trabajan en un proyecto DDD deben resistir cuidadosamente el impulso de estar tan centrados técnicamente que no puedan aceptar el enfoque comercial de la iniciativa estratégica central. Por el contrario, los desarrolladores deberían rechazar cualquier terquedad innecesaria y ser capaces de adoptar el lenguaje ubicuo que el equipo desarrolla gradualmente dentro de su contexto limitado particular (Vaughn, 2016).

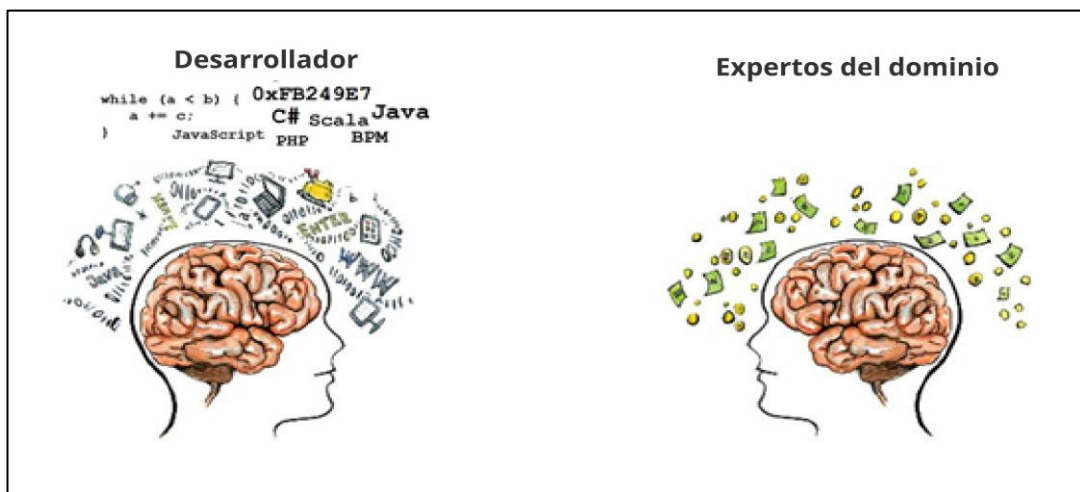


Figura 10. Representación de desarrolladores y expertos del dominio. Fuente: (Vaughn, 2016).

Test-Driven Development (TDD)

TDD es una práctica en la cual el desarrollo de encuentra dirigido por test, fue desarrollado por Kent Beck, el mismo que desarrolló Extreme Programming, por lo que se encuentra ligado con la metodología ágil. TDD consta en primera instancia escribir las pruebas, en segunda el código y por último refactorizar el mismo, esto logra generar código mantenible, de fácil desarrollo y robusto. Por lo tanto, las pruebas funcionan como una herramienta de diseño y como pruebas de regresión. Al utilizar TDD solo se hace uso de lo que se necesita en el instante, por lo que el cambio de perspectiva es fundamental. Con TDD, las pruebas tienen esencialmente tienen propósitos como impulsar el diseño, documentar el diseño y actuar como prueba de regresión, esto es lo que

convirtió a TDD en un enfoque revolucionario. (Stephens & Rosenberg, 2010)

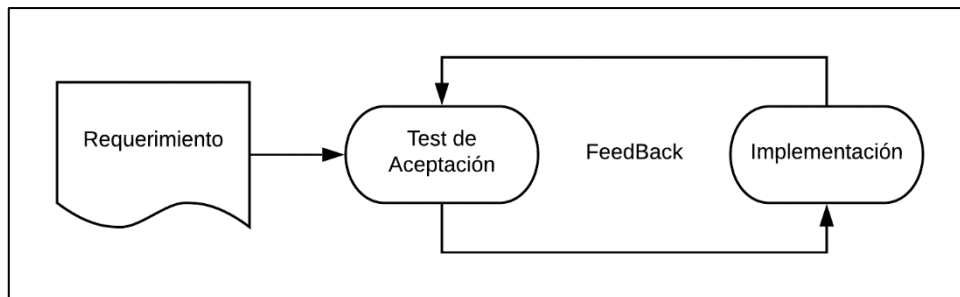


Figura 11. Ciclo de desarrollo conducido por pruebas TDD. Fuente: (Koskela, 2007).

Behavior-Driven Development (BDD)

BDD es un conjunto de prácticas que ayudan a construir y entregar software de mayor calidad, ya que se encuentra basado en prácticas como TDD y DDD. Lo más importante es que proporciona un lenguaje común que facilita la comunicación entre los miembros del equipo y las partes interesadas del negocio. BDD proviene de enseñar una forma fácil y practicar el desarrollo impulsado por pruebas. TDD está estrechamente relacionadas con una implementación particular del código, ya que se centran en el método o la función que están probando, perdiendo la imagen más amplia de los objetivos comerciales que se supone que deben implementar. Por el contrario, BDD empieza identificando los objetivos comerciales y buscando características que ayuden a cumplir estos objetivos. En colaboración con el usuario, utilizan ejemplos concretos para ilustrar estas características. Los principios de BDD ayudan a los desarrolladores a escribir código de mayor calidad, mejor probado, mejor documentado y más fácil de usar y mantener. BDD encaja con las metodologías ágiles ya que se tiende a especificar los requerimientos como historias de usuario por lo que se puede hacer uso de BDD a través del lenguaje Gherkin, este lenguaje cuenta con cinco sentencias, feature para asignar el nombre de funcionalidad, escenario para especificar la funcionalidad de la prueba, given para las precondiciones, when para las acciones a realizar, then

para validar los resultados esperados. Esto permite tener distintos escenarios de pruebas con los que se puede lograr un desarrollo optimo (Smart, 2014).

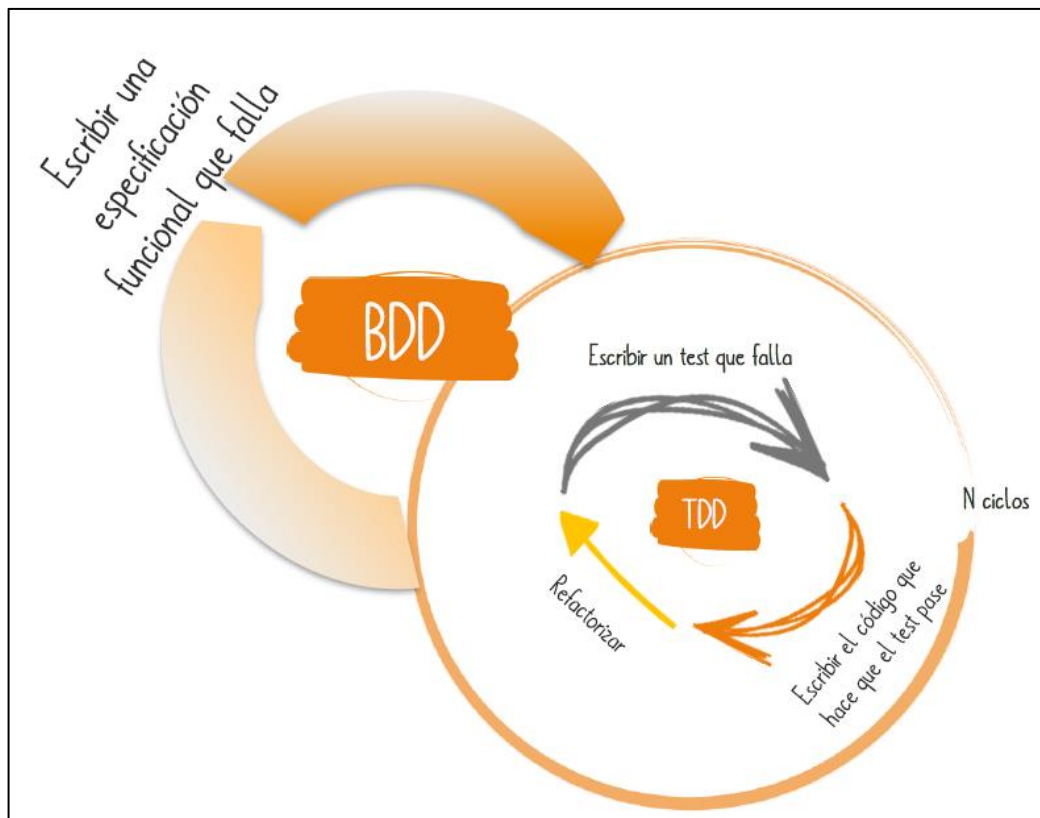


Figura 12. Desarrollo guiado por el comportamiento. Fuente: (Riol, 2017).

Representational State Transfer (REST)

Rest es un estilo de arquitectura que ha sido derivada de estilos como, capas, cliente/servidor, por lo que hace que un sistema sea confiable donde los errores puedan ser manejables, lo que diferencia a Rest de otras arquitecturas es los componentes requieren identificarse mediante direcciones globales y únicas o identificadores de identidad a las que se les llama URI, esto hace que la acuñación que se les dé a los identificadores sea independiente del contexto. Esta interfaz uniforme necesita saber el estado del recurso que ha sido recuperado, esto puede ser recuperado por media de los metadatos, para así poder saber cómo se debe analizar, procesar y comprender la representación. Los mensajes están sujetos a un protocolo de red el cual indica como deben procesarse, el protocolo HTTP define un conjunto de métodos, los cuales están

conformados por GET, PUT, DELETE, POST, los cuales son los más comúnmente utilizados. Además, de contar con un conjunto de códigos de estado los cuales están determinados por, procesamiento exitoso 2xx, redirección 3xx, falla 4xx causada por el cliente, falla 5xx causada por el servidor. Esto permite tener llevar un mejor control en la manipulación y cambios de estado de los recursos (Pautasso & Wilde, 2014).

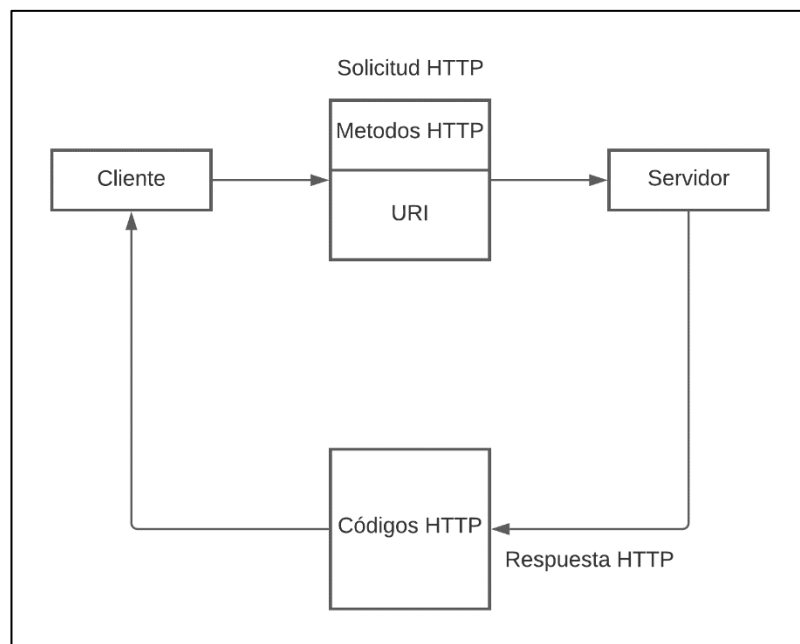


Figura 13. Diagrama REST. Fuente: (Mehta, 2014)

Diagrama de Modelo y Notación de Procesos de Negocio (BPMN)

Es una notación gráfica estándar popular para el modelamiento de procesos de extremo a extremo que proporciona coherencia y garantiza su comprensión entre las partes interesadas y entre las organizaciones. Los diagramas de BPMN utilizan símbolos básicos para visualizar los procesos de negocio, cada uno de los cuales se puede categorizar en uno de cuatro grupos: objetos de flujo (eventos, actividades y puertas de enlace), objetos de conexión (flujos de secuencia, flujos de mensajes y asociaciones), diagramas de flujo (cuadros y líneas) y artefactos (objetos de datos, grupos y anotaciones) (Lucidchart, 2021).

Elementos gráficos presentes en un BPMN

Generalmente, un diagrama BPN se dibuja de izquierda a derecha, pero podrías crear una versión vertical. Un BPMN está compuesto por elementos fundamentales que se clasifican en cuatro grupos: objetos de flujo, objetos de conexión, carriles (o piscinas) y artefactos. Cada grupo contiene símbolos que representan acciones distintas dentro de ese grupo. Hemos descrito estos términos a continuación y te mostramos cómo cada uno aparece en un BPMN.

Objetos de flujo: Son elementos que están conectados y forman un flujo de proceso.

- **Eventos:** Se muestran con círculos y representan algo que sucede. Dentro del círculo, a menudo hay íconos que indican el tipo de evento. Por ejemplo, si hay una imagen de un sobre en el círculo, eso significa que se trata de un tipo de mensaje. Además, los círculos que tienen límites sólidos se consideran eventos con interrupciones, mientras que aquellos con una línea punteada son eventos sin interrupciones. Si el círculo tiene una línea doble, se trata de un evento intermedio; y si la línea está en negrita, se refiere al fin de un evento.

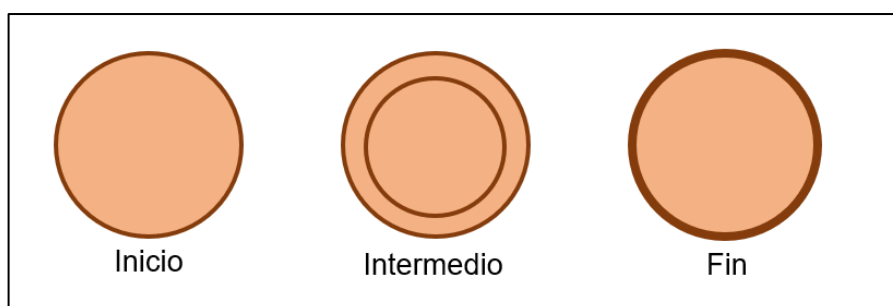


Figura 14. Objetos de flujo. Fuente: (Lucidchart, 2021).

- **Actividades:** Son procesos de negocio que se llevan a cabo, como "generar un informe". Las actividades aparecen en un BPMN como un rectángulo con bordes redondeados. Dentro de las actividades también puede haber tareas y una variedad de subprocessos.

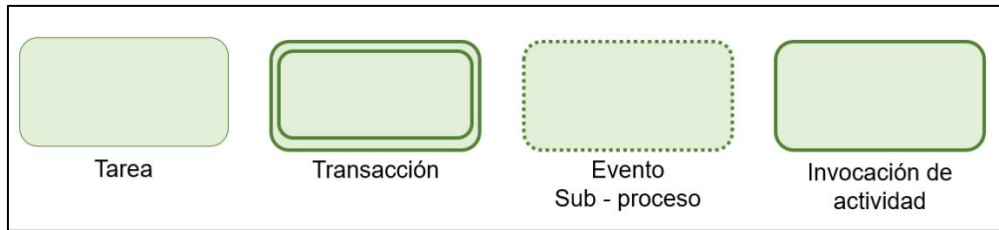


Figura 15. Actividades: (Lucidchart, 2021).

- Puertas de enlace: Si una actividad necesita pasar por diferentes flujos de procesos de negocios, debes dibujar una puerta de enlace con forma de diamante. La actividad se conecta al diamante y luego se divide señalando dos actividades distintas del flujo de proceso.

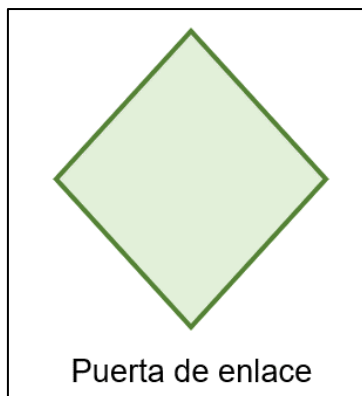


Figura 16. Puertas de enlace. Fuente: (Lucidchart, 2021).

Objetos de conexión: simbolizan cómo se conectan los objetos entre sí y representan cosas que fluyen mediante un proceso.

- Flujo de secuencia: se trata de una línea continua con una punta de flecha sólida que refleja el orden en que las actividades se llevan a cabo.

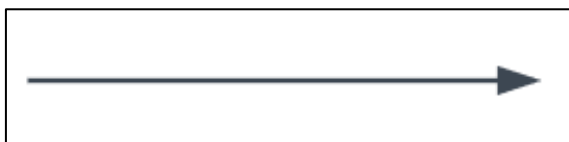


Figura 17. Flujo de secuencia. Fuente: (Lucidchart, 2021).

- Flujo de mensajes: se representa mediante una línea discontinua con un

círculo abierto al inicio de la línea y una punta de flecha abierta al final de la línea. Estos indican cuáles son los mensajes que fluyen a través de los límites organizativos.



Figura 18. Flujo de mensaje. Fuente: (Lucidchart, 2021).

- Asociación: una línea punteada que muestra relaciones entre textos y artefactos, datos, y objetos de flujo.



Figura 19. Asociación. Fuente: (Lucidchart, 2021).

Carriles/Piscina: organizan los distintos aspectos de un proceso de negocio en un diagrama de flujo multidisciplinario. Se indican mediante rectángulos grandes.

Artefactos: clasifican las actividades y pueden emplearse para incluir más información sobre un proceso en el BPMN.

- Objeto de datos: los datos requeridos para un proceso. Se parece a una hoja de papel con la esquina superior derecha doblada.

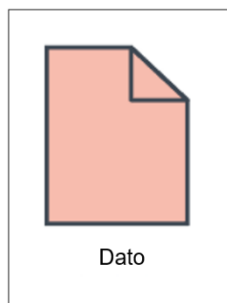


Figura 20. Objeto de datos. Fuente: (Lucidchart, 2021).

- Grupo: consiste en un rectángulo con bordes redondeados con líneas

discontinuas y agrupa diferentes actividades.

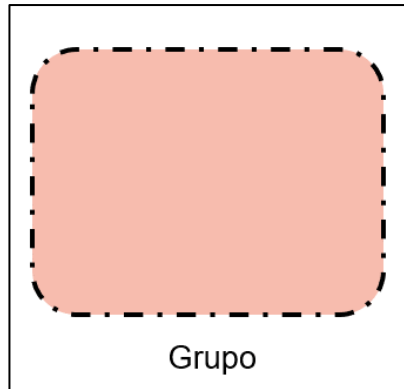


Figura 21. Grupo. Fuente: (Lucidchart, 2021).

- Anotación de texto: un texto que proporciona más información.

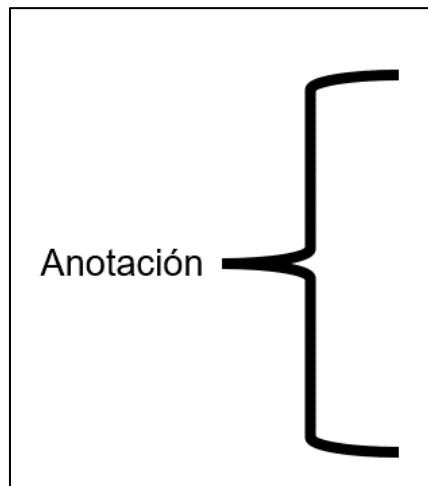


Figura 22. Anotación de texto. Fuente: (Lucidchart, 2021).

Docker

Es un programa de línea de comandos que se ejecuta en segundo plano, el cual se utiliza para resolver problemas comunes de software y simplificar su experiencia al instalar, ejecutar, publicar y eliminar software. Docker hace uso de una tecnología UNIX llamada contenedores. Cualquier software ejecutar con Docker se ejecuta dentro de un contenedor. Docker utiliza motores de contenedores existentes para proporcionar contenedores consistentes contruidos de acuerdo con las mejores prácticas. Esto pone más fuerte seguridad al alcance de todos. Con Docker, los usuarios obtienen contenedores

a un costo mucho menor. Las empresas suelen hacer uso de virtualización de hardware para así poder proporcionar aislamiento a los proyectos de software, esto permite virtualizar hardware en el que se puede instalar un sistema operativo y los programas que se requieran, con lo cual se genera una sobrecarga de recursos ya que se suele ejecutar una copia completa del sistema operativo y adicionalmente el software que se desea usar. A diferencia de las máquinas virtuales, los contenedores Docker no usan virtualización de hardware ya que los programas se ejecutan interactuando directamente con el kernel de Linux del host, esto hace que no se desperdicien recursos lo cual esta es una distinción importante de los contenedores Esta es una distinción importante, ya que permite aislar software utilizando menos recursos y logrando maximizar los beneficios (Nickoloff, 2016).

1.4. Formulación del Problema.

¿Cómo disminuir la complejidad en un proceso de transformación de una aplicación monolítica a microservicios?

1.5. Justificación e importancia del estudio.

El presente proyecto pretende atender la complejidad de transformación de un software monolítico a microservicios y el proceso que esta conlleva, esto se realiza con la finalidad de tener una metodología para dicha conversión. Esto ayudara a tener un enfoque claro de cómo se debe de realizar el proceso y que sirva como ayuda para que se lleve a cabo con la menor complejidad posible.

Con el presente proyecto se pretende abarcar desde la fase de planificación y la toma de requerimientos hasta el desarrollo y despliegue.

(Nadareishvili et al., 2016) hacen referencia que la implementación de microservicios es muy atractiva para una empresa, pero realizar esto no es

fácil, ya que las personas tienen incertidumbres frente a como esto puede afectar al sistema, estos problemas frecuentes se refieren que, si ya tienen una arquitectura a microservicios, pero no es notorio, la gestión, coordinación y control de microservicios es complejo, los microservicios no tienen contexto ni requisitos únicos.

Según (Newman, 2015) manifiesta que microservicios y su concepto de entrega continua muestran como desarrollar un software y que este entre en producción de manera más efectiva y eficiente, esto junto con las plataformas e infraestructura basadas en la nube nos permite manejar máquinas a escala. Como ejemplo de esto, tenemos a grandes organizaciones como lo son Netflix, la cual nos ha compartido maneras de crear sistemas a escala impulsado por dominios, esto permite responder rápidamente a cambios que pueden suceder en el negocio.

Como podemos corroborar los microservicios ayudan al desarrollo de software, en escalabilidad, disponibilidad, rapidez, entre otras ventajas, pero ¿cómo poder llevar una aplicación monolítica a microservicios?, este proceso debe de estar guiado por pautas, para así poder realizar una conversión adecuada y que aporte al negocio.

1.6. Hipótesis.

Mediante el desarrollo de una metodología de conversión de arquitecturas de software se podrá disminuir la complejidad en la transformación de una aplicación monolítica a microservicios

1.7. Objetivos.

1.7.1. Objetivos generales

Desarrollar una metodología para la conversión de una aplicación monolítica a microservicios basada en la nube para pequeñas

empresas.

1.7.2. Objetivos específicos

- Identificar las técnicas y herramientas más utilizadas en los microservicios desplegados en la nube.
- Desarrollar el modelo de proceso para la metodología.
- Aplicar los procesos de la metodología en un caso de estudio.
- Realizar la evaluación de la metodología mediante juicio de expertos.

II. MATERIAL Y MÉTODO

2.1. Tipo y Diseño de Investigación.

Profundizar en un tema de investigación y elegir un enfoque no sirve de nada sin tener un método y mucho menos si haber planteado un problema, plantear el problema permite estructurar la idea y las bases para el planeamiento, esto con el fin de recabar información y como analizar los enfoques, tipos y métodos de investigación a realizar (Sampieri, Collado, & Lucio, 2006).

Según (Ackoff, 1967), los problemas correctamente planteados tienen una mayor posibilidad y exactitud de obtener una solución que sea satisfactoria, lo cual pasa todo lo contrario con los problemas planteados parcialmente.

En el presente proyecto el tipo de investigación es cuantitativa – experimental dado que se realizará mediciones orientadas a la comprobación de la hipótesis, por lo que determinaremos los efectos que se pueden llegar a suceder en la variable independiente sobre la dependiente, con el objetivo de obtener resultados finales, esto con la finalidad de evaluar el proceso de conversión monolítica a microservicios.

2.2. Población y muestra.

Población

La investigación realizada en el presente proyecto la población de estudio consta de 4 métodos o procesos de conversión, estos métodos han sido indagados en artículos científicos, para que así sirvan de ejemplo a plantear una metodología acertada.

Tabla 1.

Población para el caso de estudio.

Metodo	Paper	Técnicas, metodologías	Cita
Metodo 1	Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report	DDD, Docker, Junit	(Fan & Ma, 2017)
Metodo 2	Functionality-Oriented Microservice Extraction Based on Execution Trace Clustering	Monitoreo de Ejecución, Análisis de seguimiento de ejecución, Extracción al microservicio.	(Jin, Liu, Zheng, Cui, & Cai, 2018)
Metodo 3	Efficient resources utilization by different microservices deployment models	Docker, Kubernetes, Rest Full	(Buzato et al., 2018)
Metodo 4	Performance evaluation in the migration process	Modelo de refactorización de NGInex, IBM,	(Guamán et al., 2018)

from a monolithic application to microservices Medium, Api Rest, service host

Fuente: Elaboración propia.

Muestra

La muestra no estadística ha sido determinada por conveniencia, con el fin de seleccionar los métodos o procesos, estos estarán dentro de la población previamente señalada, de los cuales tomaremos algunos elementos que han sido previamente estudiados y validados en los artículos, por lo que esto servirá para el desarrollo del proyecto de investigación.

Tabla 2.

Muestra para el caso de estudio.

Metodo	Paper	Cita	Técnicas, metodologías
Metodo 1	Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report	(Fan & Ma, 2017)	DDD, Docker, Junit
Metodo 2	Functionality-Oriented Microservice Extraction Based on Execution Trace	(Jin et al., 2018)	Monitoreo de Ejecución, Análisis de seguimiento de ejecución, Extracción al microservicio.

Metodo 3	Clustering Efficient resources utilization by different microservices deployment models	(Buzato et al., 2018)	Docker, Kubernetes, Rest Full
----------	--	-----------------------	-------------------------------------

Fuente: Elaboración propia.

2.3. Variables, Operacionalización.

Tabla 3.

Variables y operacionalización para el caso de estudio.

Variables	Dimensiones	Indicadores	Formula	Técnica e instrumento de recolección de datos
Metodología de conversión de arquitectura de software.	Evaluación de expertos	Nivel de aprobación de la propuesta metodológica	$na = \left(\frac{pp}{tp}\right) 100$	Juicio de expertos-Delphi
Complejidad de transformación de aplicación monolítica a microservicios.	Prueba empírica	Acoplamiento Cohesión	$a = \frac{nc + nd}{nt}$ $c = \frac{no}{nt}$	Análisis
		Nivel de productividad	$np = \frac{X_1 + \dots + X_n}{n}$	Cuestionario

Nivel de complejidad en el proceso de conversión metodológico	$nc = M_o$	Cuestionario
Nivel de Satisfacción del equipo involucrado en la aplicación metodológica	$ns = M_o$	Cuestionario

Fuente: Elaboración propia.

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.

Observación de campo, es una técnica que consta de observar cómo se realizan las tareas y conocer con precisión las acciones que se realizan, además de capturar cada tarea, las actividades en el contexto de su realización

Delphi, este metodo se basa en una comunicación estructurada y es iterativo de predicción, por lo que se basa en un juicio emitido por un grupo de expertos.

Cuestionarios, esta técnica se basa en preparar un documento con preguntas, estas preguntas solo serán de preguntas breves y con una respuesta concreta, este cuestionario puede ser realizado durante la entrevista, esto servirá para obtener información adicional. Para realizar el

cuestionario el analista debe tener conocimientos sobre el tema y como es factible realizarlo.

2.5. Procedimiento de análisis de datos.

VI. Metodología de conversión de arquitectura de software.

El análisis se llevará a cabo a través del método Delphi, esto para analizar los datos obtenidos de los resultados del juicio a expertos para lo que seguiremos el siguiente esquema.

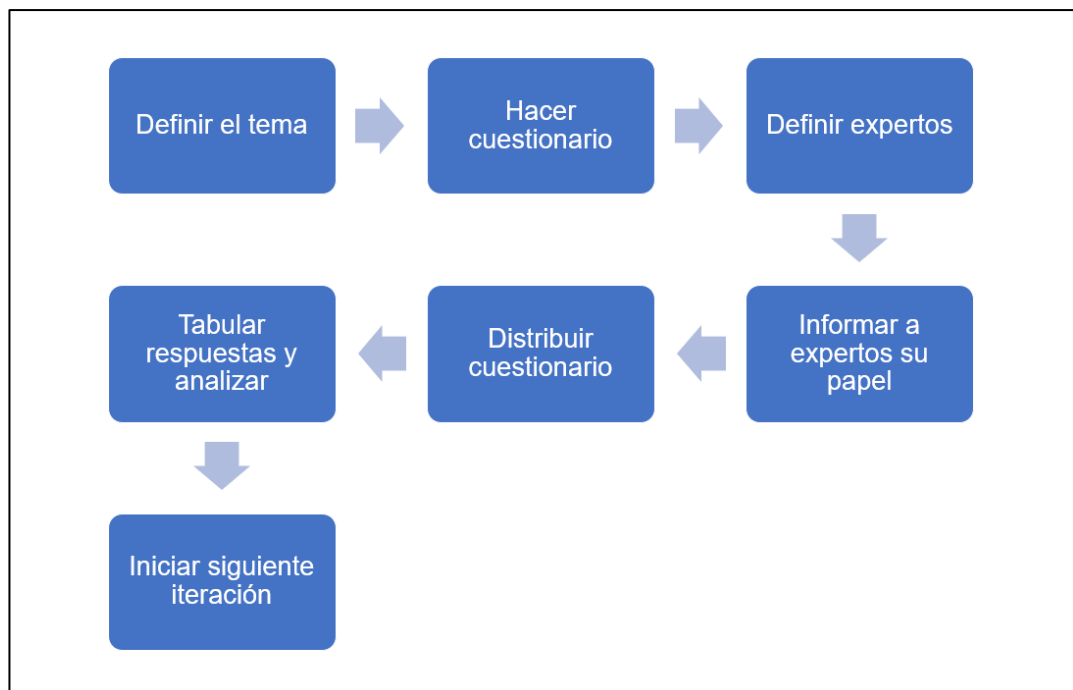


Figura 23. Iteración Delphi. Fuente: Elaboración propia.

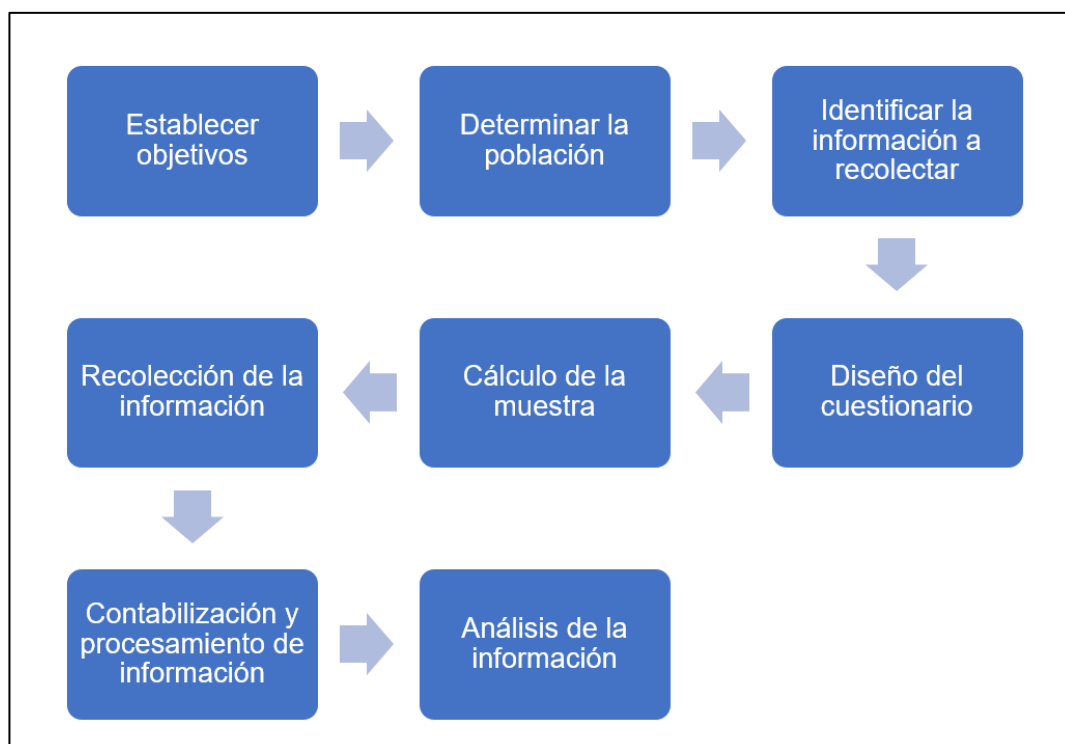


Figura 24. Formulación de Cuestionario. Fuente: Elaboración propia

Asimismo, se realizará la valorización del nivel de aprobación de la propuesta metodológica, según las iteraciones realizadas a los expertos evaluadores a través de la siguiente fórmula:

Fórmula

$$na = \left(\frac{pp}{tp} \right) 100$$

Descripción

na = Nivel de aprobación.

pp = Preguntas respondidas de manera positiva hacia la investigación según la mayoría de la decisión de los evaluadores.

tp = Total de preguntas.

El resultado del nivel de aprobación se representará en porcentajes:

- 0% - 50% = Representa una aprobación mala.
- 50% – 75% = Representa una aprobación regular.
- 75% - 100% = Representa una aprobación buena.

VD. Complejidad de transformación de aplicación monolítica a microservicios.

Además, en el proyecto de investigación tiene un enfoque cuantitativo, por lo que se realizará la evaluación de los datos obtenidos para lo cual seguiremos los siguientes pasos.

- Acoplamiento.

Según Rajendra Kharbuja, el acoplamiento es la dependencia entre el servicio y el sistema (Kharbuja, 2016).

Formula

$$a = \frac{nc + nd}{nt}$$

Descripción

a = Acoplamiento.

nc = Numero de servicios consumidos.

nd = Numero de servicios dependientes.

nt = Numero de servicios.

- Cohesión.

Según Rajendra Kharbuja, la cohesión es la relación entre las operaciones y servicios (Kharbuja, 2016).

Formula

$$c = \frac{nt}{no}$$

Descripción

c = Cohesión

no = Numero de operaciones entre servicios.

nt = Numero de servicios.

- **Nivel de productividad**

Formula

$$np = \frac{X_1 + \dots + X_n}{n}$$

Descripción

np= Nivel de productividad de los desarrolladores.

X = Productividad de los desarrolladores

n= Total de encuestados

- **Nivel de satisfacción del equipo involucrado en la aplicación metodológica**

Formula

$$ns = M_o$$

Descripción

ns= Nivel de satisfacción de todo el equipo involucrado en la aplicación de la metodología.

M_o = Nivel de satisfacción que más se repite en las encuestas.

- **Nivel de complejidad en el proceso de conversión metodológico**

Formula

$$nc = M_o$$

Descripción

nc= Nivel de complejidad en el proceso de conversión de la aplicación monolítica a microservicios.

M_o = Nivel de complejidad que más se repite en las encuestas.

2.6. Criterios éticos.

Debemos de tomar en cuenta ciertos criterios éticos como la confidencialidad para brindar la seguridad y protección de los colaboradores expresando su identidad como un anonimato o utilizando un seudónimo para los

participantes con la finalidad de que el desarrollo de la investigación se realice de la mejor manera brindando seguridad a los colaboradores, el derecho de autor representa la buena referencia y citación del material utilizado en el proceso de investigación con el fin de contribuir a los autores involucrados en el desarrollo de la investigación.

2.7. Criterios de Rigor Científico

La investigación realizada se creyó pertinente realizar un criterio de rigor científico centrándose en la ética y los valores.

Tabla 4.

Criterios de rigor científico.

Criterios	Características éticas
Fiabilidad	Analizaremos que tan fiable es la metodología propuesta.
Consistencia	Hace referencia sobre el análisis de datos, logrando aplicar las técnicas necesarias para la investigación con la finalidad de obtener como resultado una información útil y consistente para el desarrollo de la investigación
Validez	Hace referencia a el correcto análisis y evaluación de los indicadores con los cuales se pueda verificar la hipótesis.
Originalidad	Hace referencia a la correcta citación bibliográfica con cual se pueda realizar una comprobación y lograr demostrar que no se haya realizado plagio.

Fuente: Elaboración propia.

III. RESULTADOS

3.1. Resultados en Tablas y Figuras

Evaluación metodológica mediante juicio de expertos

Según la propuesta metodología MCM, se realizó una evaluación mediante juicio de expertos, permitiendo valorar la metodología MCM propuesta haciendo uso del método Delphi. Para la correcta valorización de la metodología propuesta se realizó un resumen y cuestionario, el cual fue respondido por los expertos seleccionados, los expertos fueron los siguientes:

- a) Ing. Denny Fuentes Adrianzén.
- b) Ing. Junior Eugenio Cachay Maco.
- c) Ing. Carlos Valdivia Salazar.

Según los resultados de la primera iteración realizada por los expertos evaluadores de la propuesta metodológica se realizó la valorización de las 10 preguntas remitidas en el cuestionario de evaluación de la metodología MCM.

Tabla 5.

Valoración de respuestas de la primera iteración del método Delphi.

Preguntas	Evaluador 1	Evaluador 2	Evaluador 3	Valoración
1.- Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya analizado las características de las pequeñas empresas para poder plantear la metodología MCM de acorde a dichas características?	SI	SI	SI	3/3
2.- Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya realizado un análisis de la literatura científica para extraer las características, herramientas más utilizadas en la arquitectura basada en microservicios y así poder tener un concepto general para el desarrollo de la MCM propuesta?	SI	SI	SI	3/3
3.- Según la metodología MCM propuesta. ¿Considera usted correcto que la elaboración del modelo metodológico describe mediante las actividades a realizar, un nivel de abstracción y relación comprensible para todo el proceso de la metodología MCM propuesta?	SI	SI	SI	3/3
4.- Según la metodología MCM propuesta. ¿Considera usted correcto que la elaboración de un análisis de factibilidad de migración de una arquitectura monolítica a una basada en microservicios basado en un cuestionario puede establecer un	SI	NO	SI	2/3

juicio correcto para validar la “factibilidad de migración” en la metodología MCM propuesta?

5.- Según la metodología MCM propuesta. ¿Considera usted correcto que la actividad de planeamiento de software se haga uso de una metodología ágil como Scrum, XP, con el fin de hacer mejoras continuas en el desarrollo de software?

SI **SI** **SI** **3/3**

6.- Según la metodología MCM propuesta. ¿Considera usted que para la actividad de diseño de software es correcto hacer un análisis de la tecnología actual y la tecnología a migrar a través de un cuadro comparativo para tener un concepto general de las ventajas y desventajas de las mismas?

SI **SI** **SI** **3/3**

7.- Según la metodología MCM propuesta. ¿Considera usted correcto hacer uso de un modelado con base en BPMN, con la finalidad de tener un mejor concepto del proceso a realizar?

SI **SI** **SI** **3/3**

8.- Según la metodología MCM propuesta. ¿Considera usted correcto que hacer uso del enfoque Domain Driven Design (DDD) ayuda a los involucrados en el proceso de migración a manejar un mejor lenguaje ubicuo?

SI **SI** **SI** **3/3**

9.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI retiraría/adicionaría algún elemento del modelo metodológico, para generar una metodología aplicada a una pequeña empresa?	NO	SI	NO	2/3
10.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI cree que la metodología MCM a través de sus 5 actividades disminuye la complejidad de migración de arquitectura a microservicios?	SI	SI	SI	3/3

Fuente: Elaboración propia.

Según la valoración de la primera iteración del método Delphi realizada a los expertos se ha identificado que las respuestas cumplen con el objetivo de validar la metodología MCM propuesta, esta primera valoración nos sirvió como base para reflexionar o afirmar los puntos de vista en la segunda iteración a realizar.

Tabla 6.

Valoración de respuestas de la segunda iteración del método Delphi.

Preguntas	Evaluador 1	Evaluador 2	Evaluador 3	Valoración
1.- Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya analizado las características de las pequeñas empresas para poder plantear la metodología MCM de acorde a dichas características?	SI	SI	SI	3/3
2.- Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya realizado un análisis de la literatura científica para extraer las características, herramientas más utilizadas en la arquitectura basada en microservicios y así poder tener un concepto general para el desarrollo de la MCM propuesta?	SI	SI	SI	3/3
3.- Según la metodología MCM propuesta. ¿Considera usted correcto que la elaboración del modelo metodológico describe mediante las actividades a realizar, un nivel de abstracción y relación comprensible para todo el proceso de la metodología MCM propuesta?	SI	SI	SI	3/3
4.- Según la metodología MCM propuesta. ¿Considera usted correcto que la elaboración de un análisis de factibilidad de migración de una arquitectura monolítica a una basada en microservicios basado en un cuestionario puede establecer un	SI	NO	SI	2/3

juicio correcto para validar la “factibilidad de migración” en la metodología MCM propuesta?

5.- Según la metodología MCM propuesta. ¿Considera usted correcto que la actividad de planeamiento de software se haga uso de una metodología ágil como Scrum, XP, con el fin de hacer mejoras continuas en el desarrollo de software?

SI **SI** **SI** **3/3**

6.- Según la metodología MCM propuesta. ¿Considera usted que para la actividad de diseño de software es correcto hacer un análisis de la tecnología actual y la tecnología a migrar a través de un cuadro comparativo para tener un concepto general de las ventajas y desventajas de las mismas?

SI **SI** **SI** **3/3**

7.- Según la metodología MCM propuesta. ¿Considera usted correcto hacer uso de un modelado con base en BPMN, con la finalidad de tener un mejor concepto del proceso a realizar?

SI **SI** **SI** **3/3**

8.- Según la metodología MCM propuesta. ¿Considera usted correcto que hacer uso del enfoque Domain Driven Design (DDD) ayuda a los involucrados en el proceso de migración a manejar un mejor lenguaje ubicuo?

SI **SI** **SI** **3/3**

<p>9.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI retiraría/adicionaría algún elemento del modelo metodológico, para generar una metodología aplicada a una pequeña empresa?</p>	NO	SI	NO	2/3
<p>10.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI cree que la metodología MCM a través de sus 5 actividades disminuye la complejidad de migración de arquitectura a microservicios?</p>	SI	SI	SI	3/3

La segunda iteración del método Delphi realizada a los expertos nos sirvió para afirmar los puntos de vista de los evaluadores los cuales se han mantenido, con respecto a la primera iteración.

La valorización realizada a la segunda iteración nos mostró que con respecto a la propuesta metodología remitida a los evaluadores a través del cuestionario correspondiente, en las preguntas 4 y 9 se obtuvo una respuesta global afirmativa para el investigador, los evaluadores 1 y 3 brindaron una respuesta afirmativa en base a la pregunta, pero el evaluador 2 brindó una observación de manera negativa en base a la pregunta. Asimismo, en las demás preguntas del cuestionario se obtuvo una respuesta global afirmativa de los evaluadores con respecto a la propuesta presentada.

Según lo analizado, se procedió a realizar la valorización de la aprobación de la metodología a través de la siguiente fórmula:

$$na = \left(\frac{pp}{tp}\right) 100$$

$$na = \left(\frac{10}{10}\right) 100$$

$$na = 100 \%$$

Para hallar el nivel de aprobación se reemplazó los valores correspondientes en la fórmula planteada; siendo “pp” las preguntas aprobadas y “tp” el total de preguntas. El resultado del nivel de aprobación, según la valorización fue del 100% lo que corresponde que 10/10 preguntas en mayoría de evaluadores, las cuales se respondieron afirmativamente apoyando al investigador sobre la propuesta metodológica MCM.

Acoplamiento

El análisis del acoplamiento entre los microservicios desarrollados en el caso práctico del aplicativo “GMO” se realizó haciendo uso de la fórmula empleada por Rajendra Kharbuja.

$$a = \frac{nc + nd}{nt}$$

$$a = \frac{2 + 2}{5}$$

$$a = 0.8$$

Para hallar el acoplamiento se reemplazó los valores correspondientes en la fórmula planteada; siendo “nc” el valor de 2 servicios consumidos, “nd” el valor de 2 servicios dependientes y “nt” el valor de 5 servicios totales. El resultado del acoplamiento obtenido fue de 0.8 eso indica que tiene cierto nivel de independencia, esto indica que puede afectar a ciertos servicios cuando se realicen actualizaciones.

Cohesión

El análisis de la cohesión entre los microservicios desarrollados en el caso práctico del aplicativo “GMO” se realizó haciendo uso de la fórmula empleada por Rajendra Kharbuja.

$$c = \frac{nt}{no}$$

$$c = \frac{5}{4}$$

$$c = 1.25$$

Para hallar la cohesión se reemplazó los valores correspondientes en la fórmula planteada; siendo “nt” el valor de 4 operaciones entre servicios y “n” el valor de 5 servicios totales. El resultado de la cohesión obtenida fue de 1.25 eso indica que las funcionalidades desempeñadas por los servicios están focalizadas y relacionadas entre sí brindando una responsabilidad única siendo necesarias en el sistema, con base en el enfoque DDD cada servicio debe ser cohesivo permitiendo realizar funcionalidades independientes.

Según los resultados del acoplamiento como la relación de los microservicios entre sí y la cohesión como la estrecha relación de los componentes para realizar las funcionalidades únicas de los microservicios, se pudo obtener que del caso práctico del aplicativo “GMO” los microservicios como atributos de calidad tienen un bajo acoplamiento y una alta cohesión.

$$a < c$$
$$0.8 < 1.25$$

Se reemplazo los valores correspondientes siendo “a” el acoplamiento de 0.8 y “c” la cohesión de 1.25. Favoreciendo un bajo acoplamiento podremos tener microservicios más pequeños con responsabilidades más definidas y fáciles en tender por separado, ayudando a tener un sistema más flexible en el que se pueda reconfigurar las relaciones entre las funcionalidades que se vayan a añadir en desarrollos posteriores.

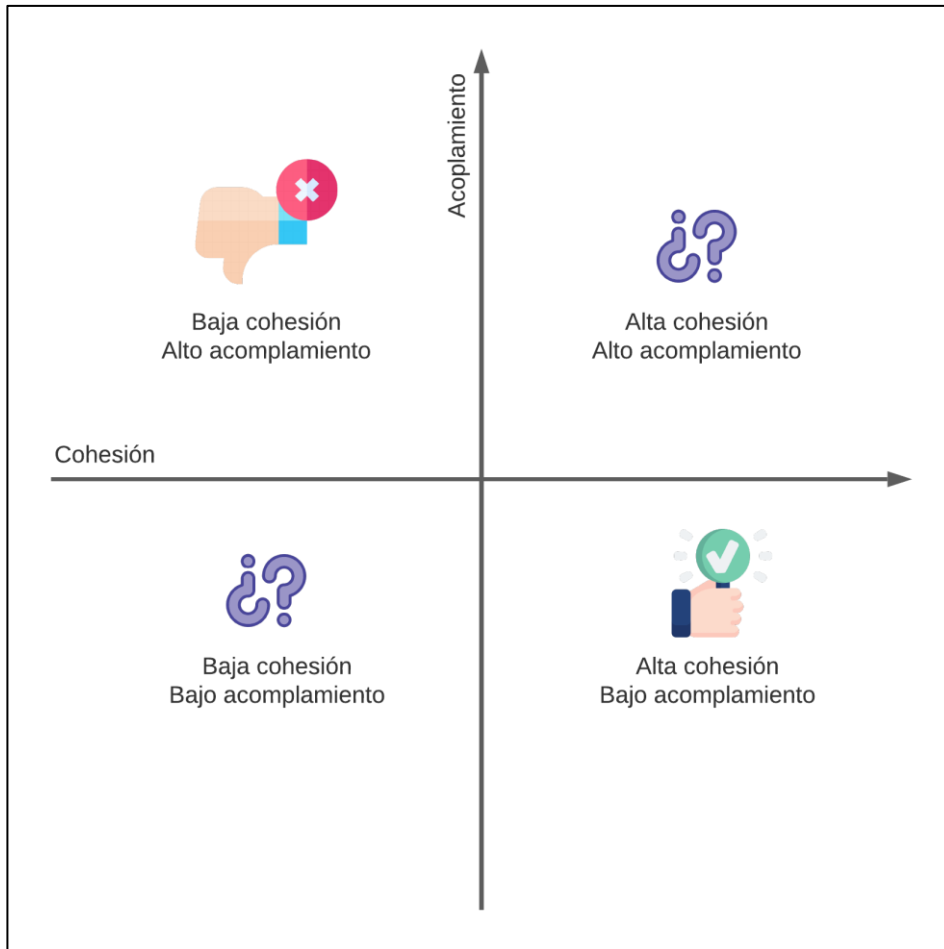


Figura 25. Diagrama de ejes de la relación entre en acoplamiento y la cohesión en los microservicios. Fuente: Elaboración propia.

Nivel de productividad

Para realizar un análisis del nivel de productividad a los integrantes involucrados en la aplicación de la metodología MCM al caso práctico del aplicativo “GMO” se realizó una encuesta la cual se muestra en el **Anexo 3.5**. Dicha encuesta permite evaluar si el nivel de productividad. La productividad se mide en cuanto al porcentaje de mejora brindado por el encuestado.

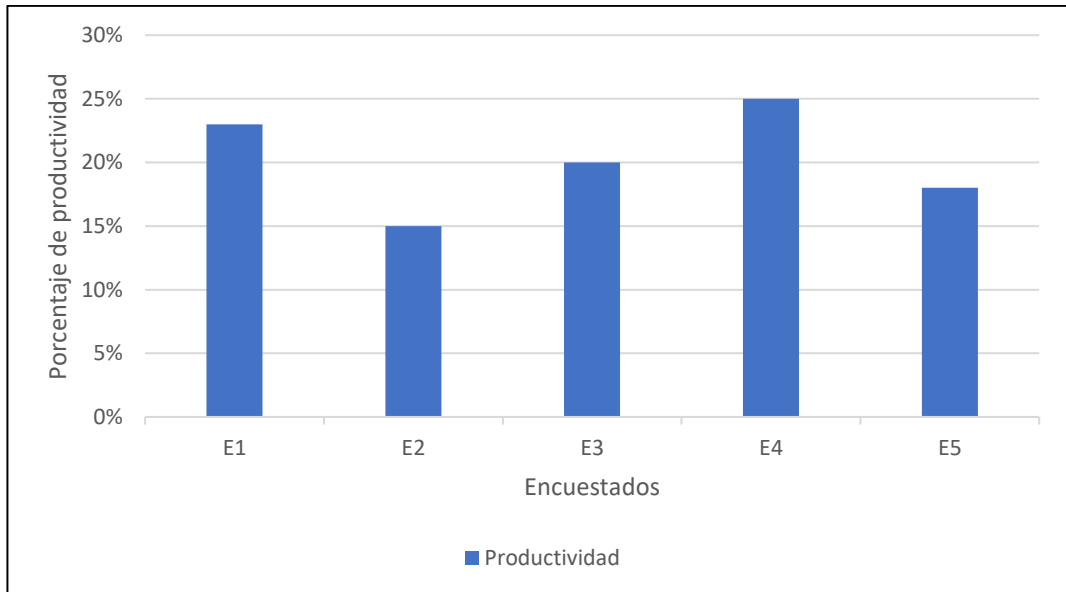


Figura 26. Evaluación del aumento de productividad del modelo propuesto. Fuente: Elaboración propia.

Según lo analizado, se procedió a obtener el promedio de mejora de la productividad a través de la siguiente fórmula:

$$np = \frac{X_1 + \dots + X_n}{n}$$

$$np = \frac{18 + 25 + 20 + 15 + 23}{5}$$

$$np = 20,2\%$$

Para hallar el nivel de aprobación se reemplazó los valores correspondientes en la fórmula planteada; siendo “x” el porcentaje de productividad y “n” el total de encuestados. El resultado del promedio de mejora de la productividad en la implementación del MCM es de 20.2% con respecto a la aplicación monolítica GMO.

Nivel de satisfacción del equipo involucrado en la aplicación metodológica

Para realizar un análisis del nivel de satisfacción a los integrantes involucrados en la aplicación de la metodología MCM al caso práctico del aplicativo “GMO” se realizó una encuesta la cual se muestra en el **Anexo**

3.5. Dicha encuesta permite evaluar si el nivel de satisfacción del equipo. La satisfacción se mide en cinco escalas de valor:

1. Muy insatisfecho
2. Insatisfecho
3. Regular
4. Satisfecho
5. Muy satisfecho

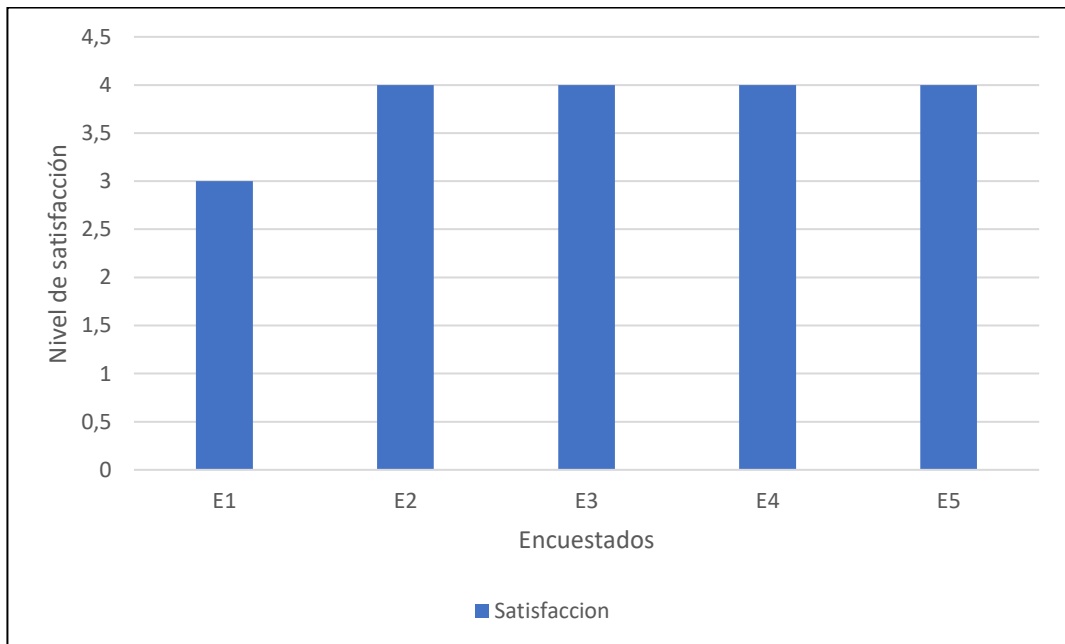


Figura 27. Evaluación de la satisfacción durante el proceso de la MCM. Fuente: Elaboración propia.

Según lo analizado, se procedió a obtener la moda para obtener el nivel de satisfacción a través de la siguiente fórmula:

$$ns = M_o$$

$$ns = 4$$

El resultado obtenido sobre la satisfacción del equipo es de 4, lo cual indica que los involucrados en el proceso de la MCM se encuentran satisfechos con el proceso y resultado.

Nivel de complejidad en el proceso de conversión metodológico

Para realizar un análisis del nivel de complejidad a los integrantes involucrados en la aplicación de la metodología MCM al caso práctico del aplicativo “GMO” se realizó una encuesta la cual se muestra en el **Anexo 3.5**. Dicha encuesta permite evaluar si el nivel de complejidad que experimentó el equipo. La complejidad se mide en cinco escalas de valor:

1. Muy baja
2. Baja
3. Regular
4. Alta
5. Muy alta

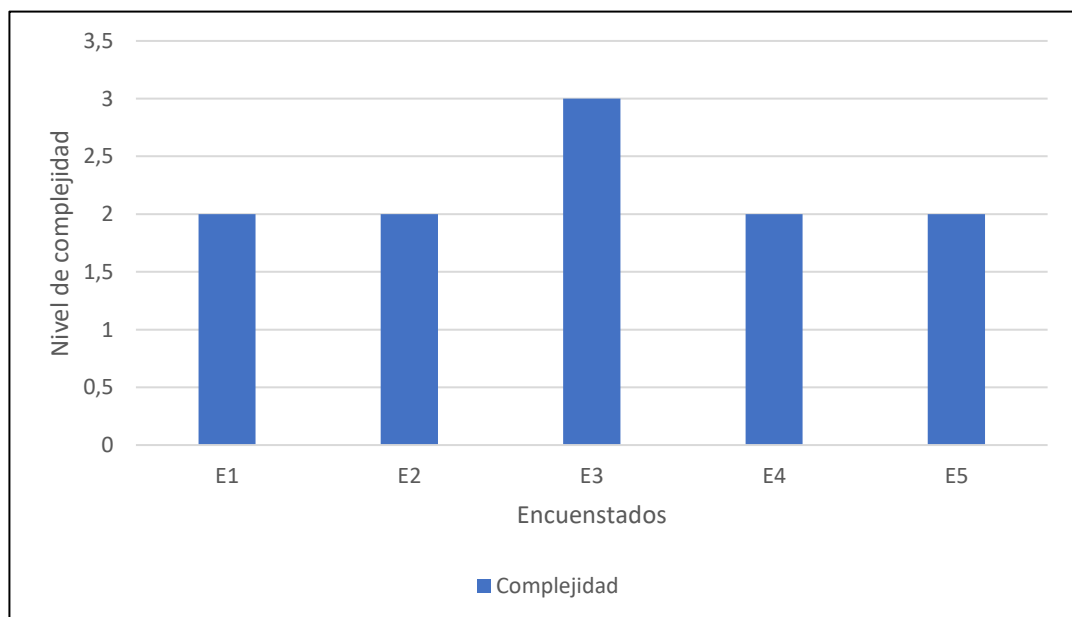


Figura 28. Evaluación de la complejidad durante el proceso de la MCM. Fuente: Elaboración propia.

Según lo analizado, se procedió a obtener la moda para obtener el nivel de complejidad a través de la siguiente formula:

$$nc = M_o$$

$$nc = 2$$

El resultado obtenido sobre la complejidad del proceso MCM es de 2, lo cual indica que el equipo tuvo una complejidad baja en el proceso de implementación de la metodología MCM en el caso práctico sobre el aplicativo “GMO”.

3.2. Discusión de Resultados

Los resultados que se obtuvieron al desarrollar la propuesta metodológica se analizaron para tener un mejor contexto y comprensión de la investigación. El análisis de la literatura científica que se realizó nos mostró ciertas técnicas y herramientas, tales como DDD, Rest, Docker Aws. Sin embargo, cuando se aplicó la metodología en un caso de estudio se tomó referencia las técnicas y herramientas obtenidas, pero no todas cumplieron con el contexto del negocio, en el servicio cloud se optó como proveedor Digital Ocean porque era el que más se adaptaba a sus necesidades.

De la misma manera el análisis de la MCM propuesta por un grupo de expertos permitió saber las opiniones de profesionales con experiencia en el campo de TI, permitiendo hacer mejoras sobre la metodología propuesta, saber sus opiniones y poder validar la hipótesis planteada en la investigación, favoreciendo el análisis conceptual del desarrollo de la MCM.

Luego de haber aplicado la metodología en un caso de estudio práctico, se obtuvieron resultados interesantes sobre el acoplamiento, cohesión, productividad, satisfacción y complejidad del proceso de la MCM. Dichos resultados nos indicaron que el acoplamiento de los módulos de “Control de ingreso y salida” los cuales son “Trabajador observado”, “Punto de entrada”, “Tipo de punto de entrada”, haciendo uso del enfoque DDD propuesto se desacoplaron en 5 microservicios, los cuales según la fórmula propuesta por Rajendra Kharbuja para valorizar el acoplamiento de los servicios en un sistema nos dio un valor de 0.8 lo cual indicaba que el nivel de independencia de los microservicios extraídos no afectaría de manera negativa. Sin

embargo, cuando haya actualizaciones pueden afectar ciertos servicios, pero ello también depende del dominio del negocio y el contexto de las mejoras sobre el sistema. El nivel de cohesión de los servicios tiene un valor de 1.25 lo que indica que las funcionalidades del sistema son independientes, esto no quiere decir que siempre sea así, influyen las características nuevas o incorporación de nuevos microservicios, dichas características o nuevos microservicios influyen en los valores de acoplamiento y cohesión.

El análisis del nivel de productividad, satisfacción y complejidad según la encuesta realizada a los integrantes involucrados en el proceso de la metodología MCM permitió tener un mejor contexto del proceso realizado en el caso práctico de la MCM. Según el nivel de productividad, el promedio del aumento de productividad con la metodología fue de un 20.2% esto se debe a que los desarrolladores involucrados en el proyecto podían fragmentar las actividades designadas, cuando se añadía una característica a la aplicación monolítica GMO se tenía que descargar todo el paquete de código, la misma persona tenía que diseñar las vistas, la lógica de negocio y los procedimientos almacenados. Sin embargo, en la MSA eso no es necesario ya que solo una persona puede desarrollar las vistas, mientras otra persona desarrolla el microservicio y otra persona realiza los procedimientos almacenados en paralelo según las historias de usuario y al manejar un estándar y lenguaje ubicuo la integración de estas características fue más rápida y productiva. Asimismo, el nivel de satisfacción de los involucrados fue "Bueno", esto se debió a que las historias de usuario, el enfoque DDD, los estándares de desarrollo permitieron una comunicación fluida entre todos los involucrados con el proceso. Al analizar el nivel de complejidad del proceso MCM por parte de los involucrados del proceso fue "Simple", esto quiere decir que entendieron todo el proceso de la MCM, al principio comprender la MCM no fue fácil, pero con las herramientas adecuadas y contexto del dominio del negocio se llegó a reducir la complejidad del proceso.

La MCM se aplicó de manera correcta en el caso práctico, obteniendo buenos resultados, esto no significa que siempre sea así, depende mucho del dominio del negocio, la magnitud de la migración y del equipo involucrado en el proceso., pero eso ya lo veremos en futuras investigaciones.

3.3. Aporte práctico

Técnicas y herramientas más utilizadas en los microservicios.

Para llevar a cabo la formulación de la metodología de conversión de una aplicación monolítica a microservicios para pequeñas empresas desplegable en la nube, la cual será llamada “Metodología de Conversión a Microservicios” (MCM). Se realizó un análisis de la literatura científica sobre las técnicas y herramientas más utilizadas en los microservicios desplegados en la nube, ello se basó en el método propuesto por (DeFranco & Laplante, 2017). La adaptación de método se basó en el plan de investigación, planteamiento de interrogantes, protocolos de revisión y validación del proceso de revisión de la literatura científica. En el procedimiento se identificó, seleccionó, evaluó, extrajo y se sintetizó los resultados de la búsqueda. Asimismo, como parte final se validó el informe. El estudio fue realizado a partir del año 2014, siendo el año en el que Martin Fowler define el concepto de microservicios (Lewis & Fowler, 2014) hasta el año 2021.

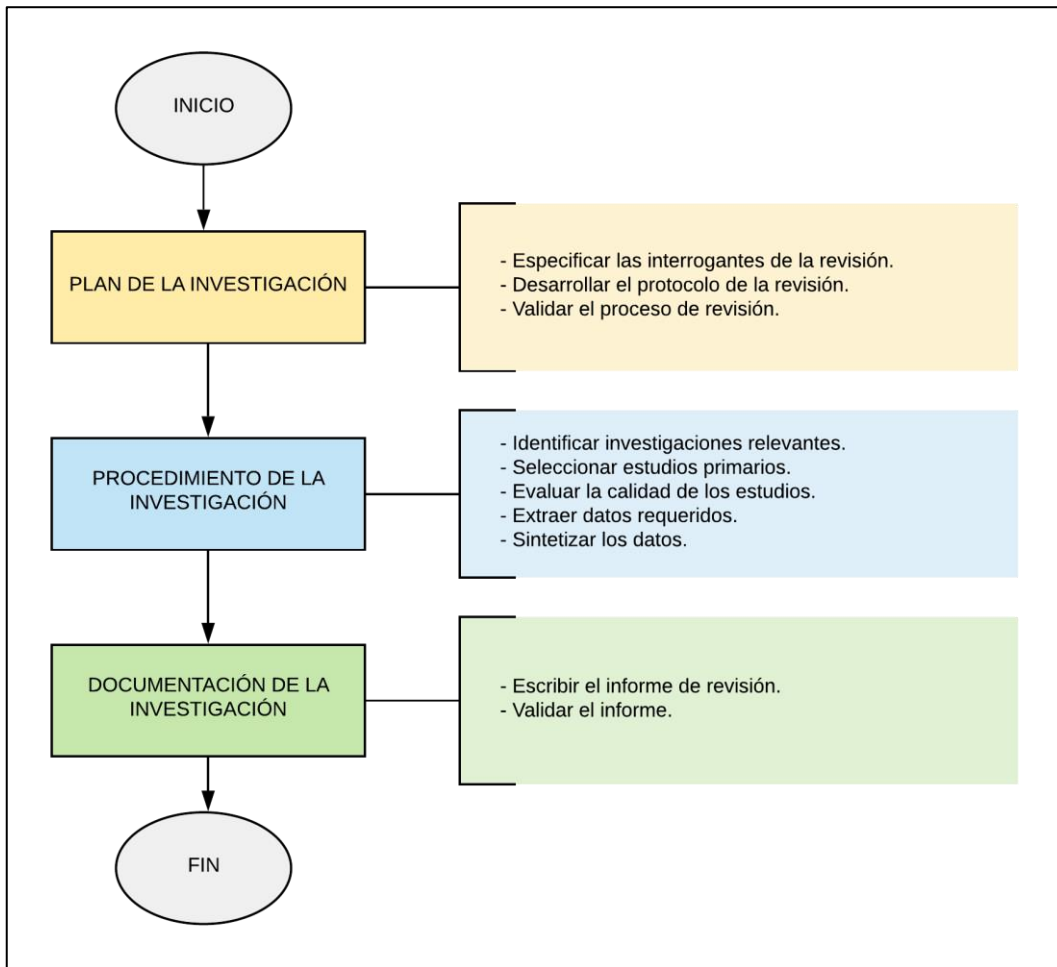


Figura 29. Método de revisión de la literatura científica. Fuente: Elaboración propia basado en (DeFranco & Laplante, 2017)

El proceso del análisis de la literatura se llevó a cabo haciendo uso de la estructura Population, Intervention, Comparison, Outcome, Context (PICOP), donde se realizó la extracción de los términos claves y sinónimos para formar la regla de búsqueda. La búsqueda se realizó en las bases de datos IEEE Xplore Digital Library, ScienceDirect y Molecular Diversity Preservation International (MDPI).

El proceso de la revisión bibliográfica se llevó a cabo especificando los siguientes pasos para garantizar la veracidad de la información:

- a) Interrogantes de investigación que el estudio responderá.
- b) Proceso que seguirá a la búsqueda de información.

- c) Condición se aplique al seleccionar los artículos a incluir en la búsqueda de la información.

Las interrogantes planteadas fueron:

PREGUNTA 1. ¿Cuánto material bibliográfico relacionado a la arquitectura basada en microservicios se ha publicado entre los años 2015 a 2019?

PREGUNTA 2. ¿Cuál es el enfoque predominante para la extracción de microservicios a una aplicación monolítica?

PREGUNTA 3. ¿Cómo se cataloga a los microservicios, arquitectura o patrón?

PREGUNTA 4. ¿Qué método, técnica, enfoques es predominante para la orquestación de microservicios?

PREGUNTA 5. ¿Cuáles son las técnicas de migración más utilizadas al momento de optar por el enfoque a microservicios?

PREGUNTA 6. ¿Qué herramienta es la más usada para encapsular los microservicios?

Se identificaron las investigaciones relevantes a través de la estructura PICOP, esta estructura sirvió para definir el ámbito de la revisión, ayudando en el proceso de selección de los términos de búsqueda. Se identificaron las propuestas más relevantes, las cuales dieron respuesta a las interrogantes planteadas. Por lo tanto, se centró en la comprensión de las técnicas y herramientas utilizadas en los microservicios desplegados en la nube.

La estructura PICOC fue la siguiente:

Population. - Los microservicios.

Intervention. - Técnicas, Herramientas utilizadas.

Comparison. - Las aplicaciones Monolíticas.

Outcome. - Arquitectura.

Context. - Arquitectura de Software.

La selección de los estudios se realizó haciendo uso de Criterios de Inclusión (CI) y los Criterios de Exclusión (CE), estos criterios se aplicaron sobre el resumen, desarrollo y conclusiones.

Criterios de Inclusión

CI1. Estudios sobre la arquitectura basada u orientada a microservicios.

CI2. Estudios donde se hayan utilizado técnicas y/o herramientas utilizadas en los microservicios.

CI3. Estudios se aborden principios de comunicación, extracción, aislamiento, orquestación, despliegue de microservicios.

CI4. Estudios se aborden conversión o migración de aplicaciones monolíticas a microservicios.

CI5. Estudios publicados desde 2014 hasta 2021.

Criterios de Exclusión

CE1. Estudios que solo hablen de conceptos la arquitectura de microservicios

CE2. Estudios de revisión sistemática de la literatura y/o comparación sobre los microservicios.

CE3. Estudios donde no se aborden técnicas, herramientas ni principios de comunicación, extracción, aislamiento, orquestación, despliegue de microservicios

Durante la búsqueda se lograron extraer 62 artículos, y aplicar el criterio de inclusión y extracción (CI/CE), se seleccionaron 48 artículos.

Tabla 7.

Resumen de las propuestas planteadas desde el 2014 - 2021.

N°	Año	Diseño (Extracción)	Comunicación	Aislamiento	Orquestación	Despliegue Cloud	Ref.
1	2015	N/E	Rest	Docker	N/E	Aws lambda	(Villamizar et al., 2015)
2	2015	N/E	Rest	Docker	N/E	N/E	(Amaral et al., 2015)
3	2016	N/E	Rest	Docker	N/E	N/E	(Kookarinrat & Temtapat, 2016)
4	2016	N/E	Rest	N/E	N/E	Aws lambda	(Villamizar et al., 2016)
5	2016	DDD	Rest	Docker	N/E	N/E	(Butzin, Golatowski, & Timmermann, 2016)
6	2016	N/E	Rest	Docker	N/E	N/E	(Linthicum, 2016)
7	2016	N/E	N/E	Docker	N/E	N/E	(Kecskemeti, Marosi, & Kertesz, 2016)
8	2016	N/E	Rest	Docker	Docker Swarm	N/E	(Salah, Zemerly, Yeun, Al-Qutayri, & Al-Hammadi, 2017)
9	2016	N/E	Rest	N/E	N/E	N/E	(Escobar et al., 2017)
10	2016	N/E	Rest	Docker	N/E	N/E	(Sadaphal & Natu, 2016)
11	2017	N/E	Rest	N/E	N/E	N/E	(Afanasev, Fedosov, Krylova, & Shorokhov, 2017)
12	2017	N/E	Rest	Docker	Kubernetes	N/E	(Sun, Li, & Memon, 2017)
13	2017	N/E	Rest	N/E	N/E	N/E	(Posadas, 2017)
14	2017	N/E	Rest	Docker	N/E	N/E	(V. Singh & Peddoju, 2017)
15	2017	N/E	Rest	N/E	N/E	N/E	(Suresh Kumar & Mallikarjuna Shastri, 2018)
16	2017	Dataflow driven (DFD)	N/E	N/E	N/E	N/E	(Chen, Li, & Li, 2018)

N°	Año	Diseño (Extracción)	Comunicación	Aislamiento	Orquestación	Despliegue Cloud	Ref.
17	2017	N/E	Rest	Docker	N/E	N/E	(Torkura, Sukmana, Cheng, & Meinel, 2017)
18	2017	DDD	Rest	Docker	N/E	N/E	(Acevedo, Gómez, & Patiño, 2017)
19	2017	N/E	Rest	N/E	N/E	N/E	(Hasselbring & Steinacker, 2017)
20	2017	DDD	Rest	Docker	Kubernetes	N/E	(Bakshi, 2017)
21	2017	N/E	Rest	N/E	N/E	N/E	(Fidge, Zimmermann, Kelly, & Barros, 2018)
22	2017	DDD	Rest	N/E	N/E	N/E	(Fan & Ma, 2017)
23	2017	N/E	Rest	N/E	N/E	N/E	(Asik & Selcuk, 2017)
24	2017	N/E	Rest	N/E	N/E	N/E	(Lin et al., 2017)
25	2017	N/E	Rest	Docker	N/E	N/E	(Zheng et al., 2018)
26	2018	BPMN	Rest	Docker	N/E	N/E	(Jahromi, Gliho, Larabi, & Brunner, 2018)
27	2018	N/E	Rest	Docker	Kubernetes	Aws lambda	(Jambunathan & Yoganathan, 2018)
28	2018	N/E	Rest	Docker	N/E	N/E	(Premchand & Choudhry, 2019)
29	2018	N/E	Rest	Docker	N/E	N/E	(A. Banati, E. kail, 2018)
30	2018	N/E	Rest	Docker	Docker Swarm	N/E	(Prachitmutita, Aittinonmongkol, Pojjanasuksakul, Supattatham, & Padungweang, 2018)
31	2018	N/E	Rest	Docker	Docker Swarm	N/E	(Bucchiarone, Dragoni, Dustdar, Larsen, & Mazzara, 2018)
32	2018	N/E	Rest	Docker	N/E	N/E	(Samal, Dubey, & Ratliff, 2018)

N°	Año	Diseño (Extracción)	Comunicación	Aislamiento	Orquestación	Despliegue Cloud	Ref.
33	2018	IBM,Medium, NGINX	Rest	Docker	N/E	N/E	(Guamán et al., 2018)
34	2018	N/E	Rest	N/E	N/E	N/E	(Garai, Adamko, & Pentek, 2018)
35	2018	N/E	Rest	Docker	N/E	N/E	(Phain & Limpiyakom, 2018)
36	2018	N/E	Rest	N/E	N/E	Aws lambda	(Qadri & Hussaan, 2019)
37	2018	N/E	Rest	Docker	Docker Swarm	N/E	(Warke et al., 2018)
38	2018	N/E	Rest	Docker	N/E	N/E	(Gan & Delimitrou, 2018)
39	2018	Model Driven Engineering approach (MISAR)	Rest	Docker	N/E	N/E	(Alshuqayran, Ali, & Evans, 2018)
40	2018	N/E	Rest	Docker	N/E	N/E	(Sarkar, Vashi, & Abdulla, 2018)
41	2018	N/E	Rest	Docker	N/E	N/E	(Ma, Fan, et al., 2018)
42	2018	N/E	Rest	N/E	N/E	N/E	(Nakazawa et al., 2018)
43	2019	N/E	Rest	Docker	N/E	N/E	(Noor et al., 2019)
44	2019	N/E	Rest	N/E	N/E	N/E	(Takeda et al., 2019)
45	2019	DDD	Rest	Docker	Docker Swarm	N/E	(Cojocar, Oprescu, & Uta, 2019)
46	2019	N/E	Rest	Docker	Kubernetes	Aws lambda	(C. Singh, Gaba, Kaur, & Kaur, 2019)
47	2019	N/E	Rest	Docker	N/E	N/E	(Sotomayor et al., 2019)
48	2019	N/E	Rest	Docker	N/E	N/E	(Grobmann & Ioannidis, 2019)

Fuente: Elaboración propia.

Los estudios realizados sobre el tema en cuestión se ordenaron de manera ascendente por el año de publicación habiéndose utilizado la regla de búsqueda correspondiente, los resultados obtenidos tienen como base la revisión realizada de artículos literarios que se relacionan con la comprensión de las técnicas y herramientas utilizadas en los microservicios desplegados en la nube.

La inclusión de los criterios para esta revisión integral de la literatura incluyó artículos de revistas que realizaron investigaciones relacionadas a las técnicas y herramientas utilizadas en los microservicios desplegados en la nube. De tal manera que se realizó una lista de verificación de evaluación de calidad.

Preguntas

PC1. ¿Se aborda el tema conceptual de microservicios?

PC2. ¿Se hace uso de alguna técnica y/o herramienta en los microservicios?

PC3. ¿Se centra en una migración de una aplicación monolítica a microservicios?

Tabla 8.

Puntuación sobre las preguntas de calidad.

Descripción	Valor
Sí	1.0
Parcialmente	0.5
No	0.0

Fuente: Elaboración propia.

Tabla 9.

Puntuación máxima y de corte para evaluar la calidad de la revisión.

Puntuación	Valor
Puntuación máxima	3.0
Puntuación de corte	1.5

Fuente: Elaboración propia.

La puntuación máxima se calcula en función del número de preguntas y la respuesta de mayor peso, Asimismo, la puntuación de corte es el valor mínimo de puntuación que se requiere para evaluar la calidad. Luego de realizar la evaluación de calidad, los resultados fueron interpretados respondiendo las preguntas planteadas en la investigación, demostrando de tal manera que las preguntas fueron desarrolladas como parte del proceso de revisión de la literatura científica. Asimismo, las técnicas y herramientas utilizadas para la migración a microservicios según el análisis de la literatura científica fueron, para la extracción (DDD), en la comunicación se usó Rest, para el aislamiento Docker, orquestación Docker Swarm y el despliegue AWS (Lambda).

Tabla 10.

Técnicas y herramientas seleccionadas por la revisión de la literatura.

Diseño (Extracción)	Comunicación	Aislamiento	Orquestación	Despliegue Cloud
Domain Driven Design (DDD)	Rest	Docker	Docker Swarm	AWS Lambda

Fuente: Elaboración propia.

Modelo de proceso de la metodología.

Para llevar a cabo el desarrollo de la metodología de conversión de una aplicación monolítica a microservicios se deben de seguir ciertos pasos que se ilustran en la siguiente imagen, a esta metodología se ha denominado con el nombre de Metodología de Conversión a Microservicios (MCM).

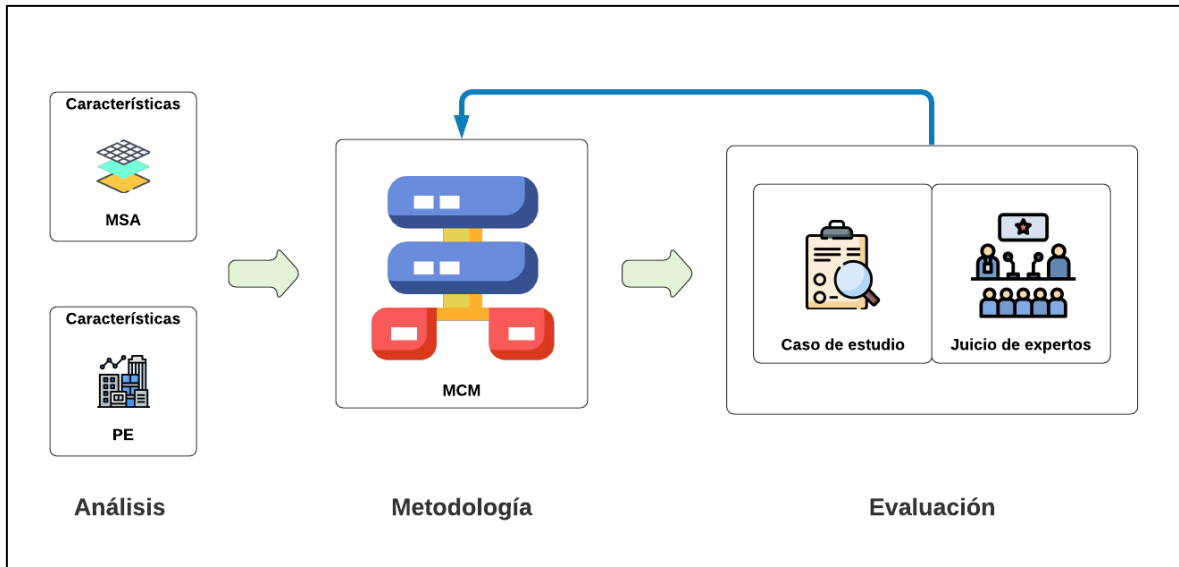


Figura 30. Proceso para el modelado de la metodología. Fuente: Elaboración propia.

El MCM se planteó para lograr la conversión de una aplicación monolítica a microservicios. Esto con el fin de lograr un mejor desarrollo de software adaptable al negocio brindando ventajas para el software como o son: escalabilidad, disponibilidad, rendimiento, versatilidad, entre otras. Por lo que, consta de tres etapas.

Análisis, en esta fase se realizó el análisis de las pequeñas empresas y/o organizaciones con el fin de identificar las características que son más relevantes para realizar una migración de su arquitectura de software, para así poder establecer criterios que brinden referencia para el buen desarrollo de la metodología a proponer.

Metodología de microservicios, en esta fase se elabora una secuencia lógica y elaborada de pasos a seguir para la construcción del modelo de migración de arquitectura monolítica a microservicios, la metodología posee tres enfoques para tener un mejor lineamiento hacia las pequeñas empresas, el enfoque de análisis del negocio se realiza con la finalidad de contar con una probabilidad de éxito alta en cuanto al desarrollo de los proyectos, para así poder entender mejor la estructura del negocio y el enfoque de desarrollo ágil el cual permita de manera colaborativa y planificada llegar a los objetivos planteados. Además de contar con un enfoque

de despliegue para comprender la manera en la que se entregará al cliente las nuevas funcionalidades de forma rápida ya que se requiere un alto grado de confianza sobre la arquitectura que se busca desarrollar.

Evaluación, en esta fase se consideró realizar mejoras a la metodología según el uso del modelo DELPHI, esto con el fin de validar la metodología.

Desarrollo del metodo

Análisis de las pequeñas empresas

En el Perú las pequeñas y medianas empresas (Mypes) tienden a clasificarse por las ventas realizadas anualmente, por ejemplo, la microempresa registra anualmente como máximo 150 UIT equivalentes a 630 mil soles, la pequeña empresa registra anualmente de 150UIT a 1700 UIT lo cual equivale a 630 mil soles y 7140 00 soles respectivamente. (Perú, 2019).

La industria del software y consultoría en el Perú se generó en el año 2014 aproximadamente 450 millones de dólares, desde entonces cada año el sector crece aproximadamente un 15%, lo que hace que las exportaciones de software estén lejos de países como Uruguay, Colombia o Argentina. En este periodo se logró una exportación de aproximadamente 28 millones de dólares. De 450 empresas el 85% se encontraban en el sector de las Mypes, además, el 20% de empresas del sector pertenecían a la Asociación Peruana de Desarrolladores de Software y Servicios Relacionados (APESOFT) y generaban el 80% de ingresos.(Quintos, 2016).

Un reporte realizado por la (World, 2016) determinó que en las pequeñas empresas peruanas el nivel de aprovechamiento de las TIC se encuentra en el puesto 90 de 139 países estudiados, por lo que en el Perú las PE tienen un bajo nivel de adopción de las tecnologías de información.

Las pequeñas empresas hacen uso de la tecnología de acuerdo a sus

procedimientos comerciales, esto quiere decir que adoptan la tecnología de que mejor se ajuste pensando en el presente, sin pensar en los futuros cambios que pueden suceder ni en el crecimiento de la empresa. (Nil, Deprost, Bernaert, & Poels, 2012). De igual forma el constante cambio que se evidencia a nivel de negocio, se ve reflejado en las tecnologías de la información, ya que ello debe de generar un aporte hacia el negocio y sus objetivos. Por lo que las tecnologías de la información se deben de proyectar y resolver las necesidades generadas por el cambio, esto hace que la tecnología de la información genere un aporte sustancial a la empresa. (Heyse, Bernaert, & Poels, 2012).

Para el desarrollo de la metodología de conversión de aplicación monolítica a microservicios se deben de tomar en cuenta ciertos criterios de las pequeñas empresas.

- La metodología MCM debe de aportar agilidad y adaptación al cambio de direccionamiento y/o objetivos que puede sufrir el negocio.
- La metodología de MCM debe ser sencilla de utilizar, aplicar y entender.
- La metodología de MCM deberá establecer una referencia de cómo hacer las cosas.
- La metodología de MCM deberá contribuir positivamente a la empresa a través de las TIC.

Análisis de la arquitectura de microservicios

Según (RedHat, 2019) los microservicios representan un estilo de arquitectura o un modo de desarrollar software. De la misma forma las aplicaciones desarrolladas bajo esta arquitectura se dividen en componentes más pequeños e independientes entre sí. Las aplicaciones desarrolladas monolíticamente o bajo un enfoque tradicional se compilan todo en una sola pieza (monolito). Sin embargo, los

microservicios son independientes y en conjunto llevan a cabo la misma tarea.

Los microservicios son utilizados para la construcción de software complejo, esto logra dividir la interfaz de usuario en el lado del cliente (Front-End), la lógica de negocio y el acceso a los datos por el lado del servidor (Back-End), por lo que funcionan como un conjunto para servir las solicitudes del cliente a través de mecanismos ligeros generalmente usando recursos API Rest Full, GraphQL basados en HTTP.(Fan & Ma, 2017).

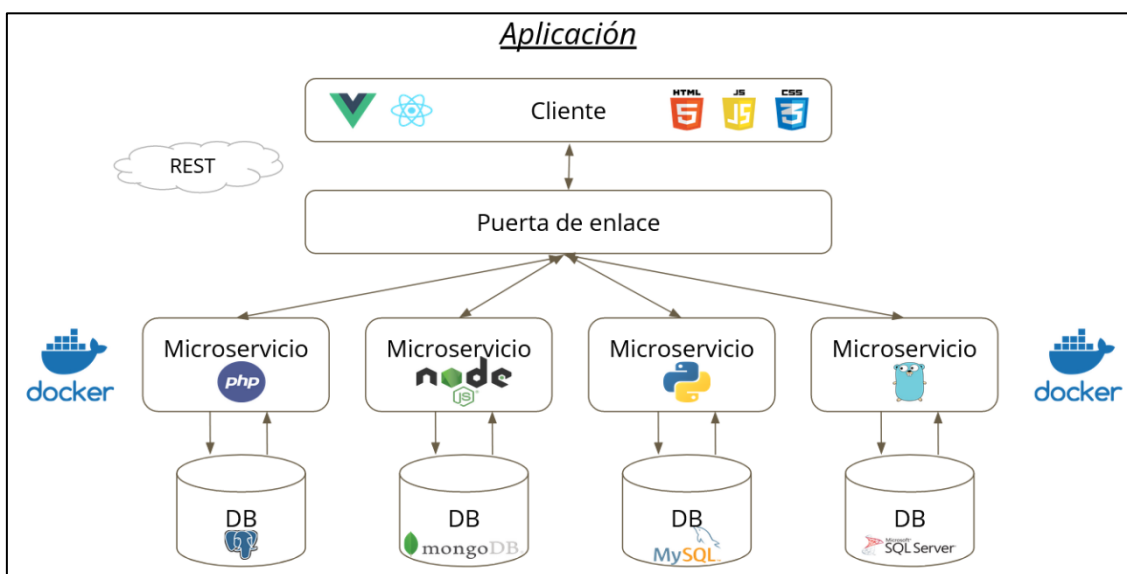


Figura 31. Representación gráfica de la arquitectura de microservicios. Fuente: Elaboración propia.

Los microservicios son pequeños e independientes, esto implica que puede ser mantenido por un único equipo reducido de desarrolladores ya que es un código base independiente, la implementación es de manera independiente por lo que pueden actualizar un servicio sin generar toda la aplicación, los servicios no pueden compartir necesariamente el mismo stack tecnológico, la comunicación se realiza mediante API's , esto permite que los detalles de la implementación interna de cada servicio se oculten frente a otros servicios. Asimismo, los microservicios brindan ventajas, agilidad para la implementación de forma independiente ya que resulta de fácil administración, equipos pequeños y centrados ya que un microservicio debe de ser lo suficientemente pequeño como para que un equipo lo mantenga, base de código pequeño para que la dependencia de código no acaben enredándose,

mezclas de tecnologías ya que se pueden elegir las tecnologías que mejor se adapten al servicio, aislamiento de errores al no estar disponible un microservicio este no interrumpe toda la aplicación, escalabilidad ya que los microservicios son independientes escalan horizontalmente los subsistemas que requieran más recursos y no toda la aplicación, aislamiento de datos ya que los microservicios son aislados es mucho más fácil realizar actualizaciones sin que toda la aplicación entre en peligro. (Azure, 2019).

Para el desarrollo de la metodología de conversión de aplicación monolítica a microservicios se deben de tomar en cuenta ciertos criterios de los microservicios.

- Los microservicios son versátiles en las tecnologías a utilizar ya que no te fuerza a trabajar con un solo stack tecnológico.
- Los microservicios permiten una agilidad y escalabilidad para el desarrollo.
- Los microservicios permiten aislar los servicios de manera independiente, eso permite controlar los errores por servicios y no se propaga en toda la aplicación.
- Los microservicios permiten mejoras rápidas y continuas, además de que pueden ser mantenidos por un grupo pequeño de desarrolladores.

Propuesta de la metodología de conversión de una aplicación monolítica a microservicios para pequeñas empresas

Tomando lo antes mencionada en el análisis de las pequeñas empresas y el análisis de microservicios se plantea la metodología la cual plantea describir los pasos a desarrollar para realizar la conversión mediante un proceso iterativo. La metodología MCM para las pequeñas empresas se basa en el análisis antes mencionado.

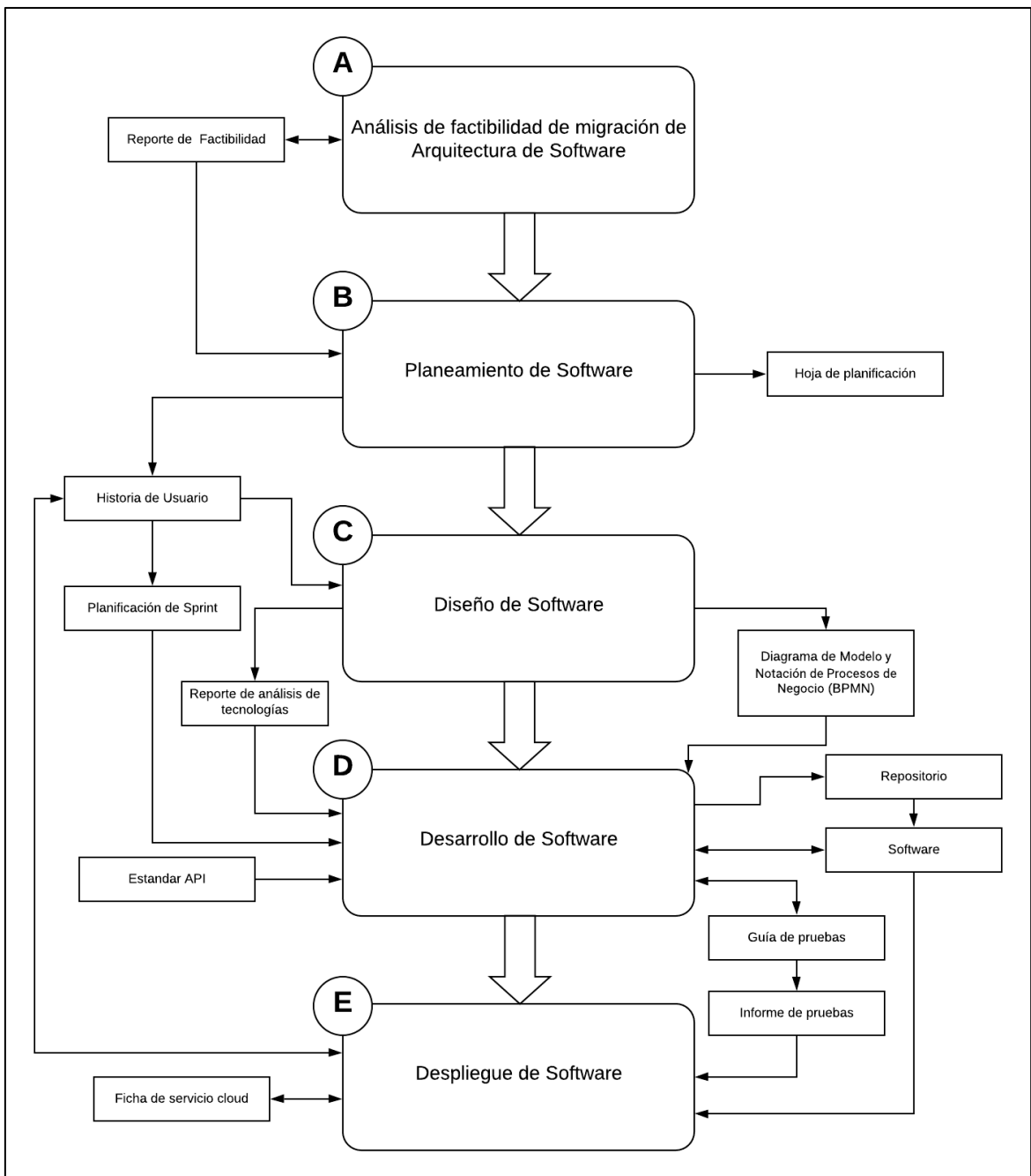


Figura 32. Metodología de conversión de aplicaciones monolíticas a microservicios desplegable en la nube para pequeñas empresas (MCM). Fuente: Elaboración propia.

Actividades de MCM:

MCM.A – Análisis de factibilidad de migración de arquitectura de software

MCM.B – Planeamiento de Software

MCM.C – Diseño de Software

MCM.D – Desarrollo de Software

MCM.E – Despliegue de Software

Roles de MCM

Cliente (C). – Dueño y financiador del proyecto

Experto de Dominio (ED). – Posee los requisitos y conoce TI y el negocio.

Líder de programación (LP). – Conoce la arquitectura y metodología a usar, además es el enlace entre el cliente, experto del dominio, debe asegurar la entrega del sistema y realizar un seguimiento durante todo el proceso de desarrollo.

Encargado de pruebas (EP). – Planifica y lleva a cabo las pruebas de software para comprobar el correcto funcionamiento.

Programador (P). – Se encarga de desarrollar el producto según la planificación del sprint realizada, además de generar un “Feedback” logrando mejorar durante las iteraciones.

MCM.A – Análisis de factibilidad de migración de arquitectura de software

La actividad de análisis de factibilidad permite:

- a) Identificar las condiciones de la arquitectura actual.
- c) Identificar los motivos para realizar la migración.

Tabla 11.

Tareas de análisis de factibilidad de migración de arquitectura de software.

Roles	Tareas	Entrada	Salida
(ED)	Realizar análisis de factibilidad de migración de	Evaluación del software	Criterio de valoración

arquitectura de software	[Realizado]
Criterio de Evaluación	

Fuente: Elaboración propia.

La actividad de análisis de factibilidad permite obtener los detalles necesarios para la toma de decisión sobre la migración de arquitectura de software. Por ello es necesario saber sobre la arquitectura en la cual se encuentra el software. Las arquitecturas tradicionales, monolíticas constan de un empaquetado donde se encuentran unidos la vista, lógica de negocio, acceso a datos y base de datos. La arquitectura de software actual debe al menos soportar los procesos de negocio, esto quiere decir que se debe de considerar los riesgos e inconvenientes que se están generando por esta arquitectura, para así poder evaluar cómo podría mejorar el rendimiento y disponibilidad del software al cambiar de arquitectura. Sin embargo, para ello se debe de evaluar la arquitectura actual, dicha evaluación involucra al Gerente de Tecnologías de la Información y se realizará mediante la siguiente tabla.

Formato de evaluación para la migración de arquitectura		
N°	Pregunta	Respuesta
1	¿De qué manera la arquitectura orientada a microservicios sigue el lineamiento de los objetivos de la empresa?	
2	¿Cuáles son los objetivos a conseguir?	
3	¿Cómo ayudará la arquitectura orientada a microservicios a alcanzar los objetivos de la empresa?	
4	¿Cuáles serían las debilidades para realizar la migración?	
5	¿De qué manera la arquitectura orientada a microservicios beneficiaría el desarrollo de software constante?	
6	¿Se tienen los recursos económicos adecuados?	
7	¿Cuál es el periodo de tiempo previsto?	

Figura 33. Formato de para evaluar la migración de arquitectura de software. Fuente: Elaboración propia

Una vez contestado la tabla de evaluación se tendrá un mejor concepto de la

arquitectura actual y en qué medida la arquitectura objetivo, en este caso la orientada a microservicios puede aportar con el fin de mejorar y lograr alcanzar los objetivos del negocio. Para ello, una vez estando en contexto de lo que pasa con la arquitectura actual se debe de realizar una valoración de valoración de esta arquitectura.

La valoración de realiza mediante tres estados, bajo, medio, alto, esto permitirá saber si la arquitectura actual es realmente fiable para el negocio, si la fiabilidad es alta significa que no es necesario de realizar una migración de arquitectura de software por lo que la arquitectura actual soporta las necesidades del cliente y del negocio. Sin embargo, si la fiabilidad es baja significa que la arquitectura de software actual no se está aportando satisfactoriamente las necesidades del cliente y del negocio, por lo que tampoco aportará positivamente al negocio, lo cual se traduce en la necesidad de cambiar de arquitectura de software, teniendo como destino la arquitectura orientada a microservicios.

Formato de valoración de la arquitectura actual				
N°	Criterio de valoración de la arquitectura de software actual	Bajo	Medio	Alto
1	Sigue el lineamiento de los objetivos de la empresa			
2	Brinda flexibilidad			
3	Brinda alta disponibilidad			
4	Permite hacer mejoras continuas eficientemente			
5	El rendimiento es el esperado			
6	Es fácilmente escalable			
7	Es versátil cuando se necesita hacer uso de nuevas tecnologías			
Total		0	0	0

Figura 34. Formato de valoración de la arquitectura de software actual. Fuente: Elaboración propia.

MCM.B – Planeamiento de Software

La actividad de planeamiento de software permite administrar el proyecto, para así lograr:

- a) Identificar las necesidades y recursos
- b) Estimar el tiempo y costo
- c) Identificar los riesgos del proyecto
- d) Realizar el análisis de requerimientos
- e) Revisar la calidad a través de los productos entregables.
- f) Crear un repositorio de proyectos para almacenar y controlar las versiones y entrega del producto.

Tabla 12.

Tareas de planeamiento de Software.

Roles	Tareas	Entrada	Salida
(ED) (LP)	Revisar el análisis de factibilidad de migración de arquitectura de software	Análisis de factibilidad para realizar la migración	Análisis de factibilidad para realizar la migración. [Revisado]
(C) (ED) (LP)	Establecer la Hoja de ruta donde se especifica el proyecto	Hoja de Ruta	Hoja de Ruta [Aceptada]
(C) (ED) (LP)	Realizar el análisis de requerimiento	Historia de Usuario	Historia de Usuario [Realizada]

Fuente: Elaboración propia.

MCM.B.1 – Hoja de Planificación

Luego de haber aplicado el análisis de factibilidad de migración de arquitectura de software y haber evaluado sus resultados, siendo que el resultado de la evaluación

indica que es factible se llevará a cabo el planeamiento, para lo cual nos basaremos en el modelo del Acta de Constitución de la guía de PMBOK, lo cual es un proceso para definir la existencia de un proyecto, sirviendo como directriz para tener una concepción clara, sencilla y formal de lo que se va a realizar, además brindar información vital y dar la autoridad para la asignación de los recursos necesarios para llevar a cabo las actividades (Institute, 2013).

Hoja de Planificación			
Nombre del proyecto		Fecha	
Responsable		Sponsor	
Financiamiento/Coste			
Descripción (¿En qué consiste?)			
Justificación (¿Por qué es importante llevarlo a cabo?)			
Roles			
Nº	Nombre Completo	Rol	
1			
2			
3			
4			
5			
Riesgos y Limitaciones			

Figura 35. Hoja de planificación. Fuente: Elaboración propia.

MCM.B.2 – Historia de usuario

Es la encargada de especificar que funcionalidades son importantes para el usuario y el desarrollo, además de especificar el tiempo estimado, prioridad, validaciones, entre otros, con el fin de ayudar a conceptualizar lo que se requiere (Mike, 2004).

La historia de usuario se debe ser implementada y descrita en un lenguaje común para el usuario, debe de estar estructurada siguiendo una narrativa del universo de discurso, cada historia debe de ser limitada ya que en el momento de la implementación se pueden surgir cambios en el negocio o esquema lo cual es bienvenido en el desarrollo ágil de software, para ello deben de ser acompañadas de conversaciones y tener la definición de las validaciones asociadas, las validaciones o criterios de validación permiten ratificar si se han recogido de manera correcta los requisitos, estas pruebas pueden ser adecuadas con TDD, BDD o ambos en conjunto, finalmente lo antes descrito servirá para comprobar si la historia de usuario ha sido completada totalmente (Menzinsky et al., 2019).

Historia de Usuario		N°:
Nombre de historia:		
Usuario:		
Prioridad de desarrollo:	Riesgo de desarrollo:	Dependencia:
Tiempo estimado:	Número de Iteración:	
Responsable:		
Descripción:		
Validación:		
Estado:		

Figura 36. Historia de usuario. Fuente: Elaboración propia.

La historia de usuario está compuesta por:

Número. – Representa el numero de la historia de usuario.

Nombre de la historia. – Representa el nombre que se le quiere dar a la historia de usuario.

Cliente. – Representa quien va a hacer uso del sistema.

Prioridad de desarrollo. – Indica el nivel de prioridad de la historia, puede estar representado con números (1,2,3,4) o indicadores verbales (Bajo, Medio Alto, Extremo).

Riesgo de desarrollo. – Representan los riesgos que conllevan el desarrollo adecuado de la historia de usuario.

Dependencia. – Representa si la historia tiene una dependencia de otra historia.

Tiempo estimado. – Representa el tiempo que le tomará al desarrollador terminar la historia de usuario.

Número de iteración. – Representa el número de iteración en la que se encuentra la historia de usuario.

Responsable. – Representa el o los responsables del desarrollo de la historia de usuario.

Descripción. – Representa la descripción de una función a realizar, dicha descripción se deberá representar en forma de diálogo. De hecho, Behavior-driven development (BDD) en un desarrollo ágil tiene un encaje significativo, ya que permite llevar a cabo un diseño basado en comportamiento, esto se puede integrar en las historias de usuario fácilmente para lograr obtener los comportamientos que se requieran, de estos comportamientos se pueden obtener los criterios de aceptación y del mismo desprender las pruebas correspondientes. Por lo que pueden ser escritas mediante el lenguaje Gherkin, el cual consta de texto con una estructura definida (Ma, Chuang, et al., 2018).

```
1 Feature: Multiplicar dos numeros.  
2  
3 Scenario: Multiplicar dos numeros positivos.  
4   Given que estoy en la intefaz  
5   When cuando ingreso los numeros 5 y 5  
6   And se solicita el resultado  
7   Then el resultado debe de ser 25  
8
```

Figura 37. Estructura del lenguaje Gherkin. Fuente: Elaboración propia.

En base a lo presentado se puede contextualizar que “Feature” representa el nombre de la funcionalidad, “Scenario” representa cada prueba que se quiera realizar para dicha funcionalidad, “Given” representa el contexto de las precondiciones, “When” representa las acciones a ejecutar, “Then” representa los resultados esperados de las validaciones. Es recomendable el comportamiento sea descrito por el cliente, para que luego el tester se asegure de que el comportamiento se encuentre bien escrito y completo. BDD ayuda a testear el sistema de la mejor forma, por lo que es deducible que funcionaria muy bien con los microservicios. BDD se puede implementar en conjunto con las historias de usuario para mejorar la conceptualización de lo que el cliente quiere y necesita. Además, de BDD se desprenden los test donde la integración con TDD es totalmente compatible para el desarrollo. (Riol, 2017).

Validaciones. – Representa las validaciones a realizar en la historia de usuario para que se desarrolle correctamente. Asimismo, si se hace uso del lenguaje Gherkin las validaciones ya se encuentran explicitas.

Estado. – Describe el estado a nivel descriptivo o porcentual el desarrollo de la historia de usuario.

MCM.B.3 – Planificación de Sprint

El Sprint es un ciclo o iteración que permite tener un ritmo y tiempo prefijado, esperando una duración aproximada de 4 semanas. Cada Sprint debe de generar un entregable del producto (Menzinsky et al., 2019). La planificación del sprint se

debe de realizar o tener como guía el siguiente formato.

Planificación de Sprint				
Sprint	Historia	Nombre de la Historia de Usuario	Nivel de prioridad	Duración estimada
S001	H001			
	H002			
	H003			
	H004			
	H005			
	...			
Total				

Figura 38. Planificación de sprint. Fuente: Elaboración propia.

MCM.B.4 – Seguimiento Diario

El seguimiento diario se realiza mediante reuniones, estas reuniones se deben de realizar en el mismo lugar y a la misma hora durante cada uno de los días, se deben realizar usualmente en las mañanas, esto se debe realizar para tener el contexto para el resto de día, el tiempo aproximado de duración es de 15 -20 min. La reunión debe de realizarse con todos los miembros del proyecto además de tocar puntos clave, con el fin de identificar las cosas que se están realizando de la manera correcta y las que generan dificultad. Por ello, el siguiente formato consiste en tener un correcto seguimiento diario.

Ficha de seguimiento diario		
Encargado:		H. Inicio – H. Fin y Fecha:
Nº de Historia de Usuario	Observación	Realizado por
Observaciones Adicionales		
Observación	Realizado por	

Figura 39. Seguimiento diario. Fuente: Elaboración propia.

La ficha de seguimiento diario está compuesta por:

Encargado. – Es la persona encargada de realizar la reunión.

H. Inicio – H. Fin y Fecha. – Describe la hora de inicio y final de la reunión, además de contar con la fecha.

Número de Historia de Usuario. – Hace referencia a la historia con la que se tiene algún inconveniente.

Observación. – Describe los inconvenientes que se generaron al desarrollar la historia de usuario.

Realizado por. – Nombre de la persona que generó la observación.

Observaciones Adicionales. – Describe los inconvenientes que se encuentran fuera del desarrollo de la historia de usuario, por ejemplo, fallas con el computador, el internet, entre otros.

MCM.C. – Diseño de Software

La actividad de diseño de software permite lograr:

- a) Realizar un análisis de tecnología
- b) Realizar un diseño del análisis de requerimientos

Tabla 13.

Actividades de diseño de software.

Roles	Tareas	Entrada	Salida
(LP)	Realizar un análisis de tecnologías		Análisis de Tecnologías [Realizado]
(LP) (P)	Realizar los diagramas de Modelo y Notación de Procesos de Negocio (BPMN)	Historia de Usuario [Revisado]	Diagramas BPMN [Realizado]

Fuente: Elaboración propia.

El diseño de software proporciona la estructura de cómo se debe cumplir con las funciones requeridas y como cumplir con la calidad deseada, identificando las fases, actividades, tareas y la relación entre sí, por lo que estos productos colectivamente se denomina diseño de software, estos productos varían por factores como la perspectiva, lenguaje, propósito, logrando una documentación descriptiva de tipo caja negra (Otero, 2012).

Análisis de Tecnologías

Cuando se realiza la migración de arquitectura a microservicios se puede reevaluar el uso de las tecnologías que hace uso el software o las que se necesitan para lograr un mejor desarrollo, en base a que el concepto de microservicios es versátil en cuanto a la integración de tecnologías, por lo que se pueden comparar las tecnologías con el fin de mejorar el software, asimismo se puede actualizar las tecnologías a utilizar por lo que un cuadro comparativo brinda un mayor enfoque y contexto, para que en base a los resultados de la evaluación de tecnologías se brinde las acciones correspondientes.

Comparación \ Tecnología	Actual Nombre de la tecnología actual	Propuesta Nombre de la tecnología propuesta
Definición ¿Qué es?		
Uso ¿En qué contexto se usa?		
Plataforma ¿Sobre qué sistema operativo se ejecuta y es compatible?		
Integración ¿Cómo y sobre que se integra?		
Licencia Describe qué tipo de licencia utiliza		
Comunidad Describe qué contribución genera la comunidad		
Peticiones Describe cuántas peticiones soporta por segundo		
Tiempo de respuesta Describe cuánto es el tiempo de respuesta por segundo		
Velocidad Describe Cuánto es el tiempo de ejecución		

Figura 40. Análisis de tecnologías. Fuente: Elaboración propia.

Domain-Driven Design

DDD permite hacer uso de una estructura práctica que ayudan en enfocar y acelerar desarrollos complejos para ayudar en la toma de decisiones. Para ello se debe tener un idioma omnipresente dentro del contexto, los beneficios son

notables al tener convenciones de modelado rigurosas con base en el dominio, el cual se representa mediante una esfera de conocimiento, influencia o actividad. (Evans, 2014). Los expertos del dominio y desarrolladores se deben encontrar en igualdad de condiciones teniendo sentido para el negocio y el software que se produce, no solo para los desarrolladores, por lo que se debe convertir en un equipo unido, esto puede lograr comprender más del negocio, ya que todos deben contribuir en las discusiones logrando comprender el software y el negocio, esto significa que el equipo en conjunto desarrolla un lenguaje común y compartido entre todos los miembros. El diseño es cómo funciona. Conocer el mejor diseño se obtiene mediante modelos experimentales rápidos que utilizan un proceso de descubrimiento ágil. DDD proporciona técnicas sólidas de desarrollo de software que abordan el diseño estratégico y táctico. El diseño estratégico nos ayuda a comprender cuáles son las inversiones en software más importantes que hacer, qué activos de software existentes aprovechar para llegar más rápido y seguro, y quién debe participar. El diseño táctico nos ayuda a crear el modelo único y elegante de una solución utilizando bloques de construcción de software probados y probados por el tiempo. DDD comúnmente es utilizado cuando la lógica es compleja a un nivel empresarial, mas no cuando se tiene una complejidad técnica o una aplicación simple. (Vaughn, 2013).

De tal manera que al diseño técnico y estratégicos que muestra Evans como definición, se le extraerá ciertos patrones con los cuales se pueda mantener una comprensión y sencillez la cual es la que se busca, para que según lo seleccionado se pueda contextualizar el dominio de negocio y tener una concepción de los pasos a tener como referencia para lograr unas mejores prácticas, con lo que cabe resaltar que la selección se realizó en base a las necesidades básicas para el dominio. Asimismo, según el enfoque DDD se buscará desacoplar la aplicación monolítica a los microservicios respectivos, según el análisis del dominio del negocio.

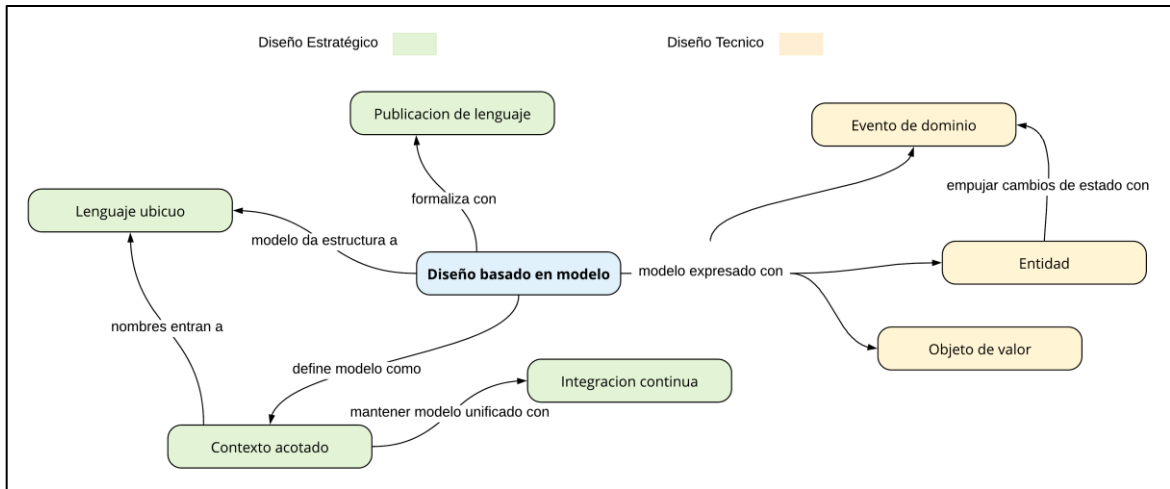


Figura 41. Descripción de lenguaje de patrones seleccionados para el nivel de diseño. Fuente: (Evans, 2014).

Diagrama de Modelo y Notación de Procesos de Negocio (BPMN)

BPMN permite tener un concepto gráfico del proceso del dominio del negocio proporcionando coherencia y garantizando su comprensión haciendo usos de símbolos básicos, los cuales representan distintas acciones y permitiendo visualizar los procesos de negocio (Lucidchart, 2021).

MCM.D – Desarrollo de Software

Esta actividad comprende el desarrollo del software, por ello se debe:

- a) Realizar la revisión el equipo de trabajo.
- b) Comprender el estándar de desarrollo.
- c) Comprender las historias de usuario y los modelos UML.
- d) Comprender y aplicar los casos de prueba.

Tabla 14.

Tareas de desarrollo de software.

Roles	Tareas	Entrada	Salida
(LP) (P)	Comprender y asignar las tareas de acuerdo a la planificación del sprint a realizar	Planificación de Sprint [Revisado]	
(LP)	Revisar el análisis de tecnologías	Historia de usuario [Revisado]	
(LP)	Crear el repositorio del proyecto donde se almacenará el software	Análisis de Tecnología [Revisado]	Repositorio [Creado]
(LP) (P)	Construir o actualizar el software siguiendo los estándares de programación.	Historia de usuario [Revisado]	Software
(EP)	Realizar casos de prueba al software desarrollado de acuerdo con la especificación de requisitos. El cliente proporciona datos de prueba, si es necesario.	Diagrama UML [Revisado]	Repositorio [Actualizado]
		Historia de usuario [Validado]	Guía de pruebas [Realizado]
		Software	Informe de Pruebas [Realizado]
(P)	Corregir los casos de prueba según los defectos encontrados	Guía de pruebas [Revisado]	Software [Actualizado]
		Informe de Pruebas [Revisado]	Repositorio [Actualizado]
		Software	

Fuente: Elaboración propia.

Repositorio del Proyecto y sistema de control de versiones (VCS)

El desarrollo de software ha estado dividido en versiones y debe de estar almacenado en un solo lugar. El VCS gestiona la historia del proyecto, ofreciendo la posibilidad de recuperar una versión anterior o permitiendo el movimiento entre las versiones con la finalidad de modificar, mejorar, revisar el software desarrollado. Git es un sistema de control de versiones de código libre, permite gestionar los cambios realizados durante el desarrollo, además de brindar información de las personas que realizaron los las modificaciones y el motivo (Straub & Ben, 2014).



Figura 42. Git. Fuente: (Git, 2021)

El almacenamiento del proyecto a través de una plataforma como lo es Github, Gitlab, Bitbucket permiten el almacenamiento del proyecto de manera remota, ofreciendo la posibilidad de trabajar de manera colativa y realizar un seguimiento del proyecto. Los comandos básicos a utilizar son los siguientes:

Crear un repositorio en un directorio específico.

```
git init
```

Añadir todos los archivos del directorio.

```
git add .
```

Confirmar cambios asociando un mensaje descriptivo.

```
git commit -m "mensaje"
```

Subir los cambios a un servidor remoto.

```
git push origin "nombre rama"
```

Descargar los cambios disponibles de un servidor remoto.

```
git pull origin master
```

Desarrollo basado en estándares. – El desarrollo haciendo uso de estándares promueve el entendimiento del equipo, logrando facilitar la recodificación del software. Además, los estándares y el uso del lenguaje ubicuo ayudan a mejorar la comunicación y el desarrollo (Vaughn, 2016).

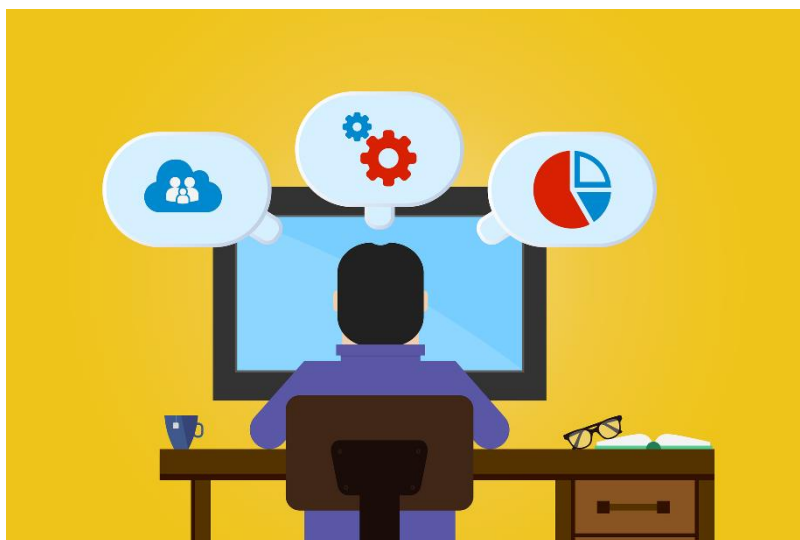


Figura 43. Representación de estándares. Fuente: Elaboración propia.

El desarrollo de API's es muy común para la comunicación entre microservicios, por ello es óptimo contar con un estándar de representación donde se pueda llegar a un entendimiento entre todo el equipo, para ello se presenta la siguiente ficha:

Módulo. – Nombre del módulo al que pertenece

Funcionalidad. – Describe la funcionalidad que realiza la API

API. – Describe la URI a consumir

Método. – Describe el verbo a utilizar (POST, GET, PUT, DELETE)

Entrada. – Describe el formato y los datos que necesita la API para realizar una lógica.

Salida. – Describe el formato y los datos ya procesados.

Ficha de API'S					
Modulo	Funcionalidad	API	Metodo	Entrada	Salida

Figura 44. Ficha de Apis. Fuente: Elaboración propia.

API Gateway / Pueta de enlace. – Es la encargada de ser el único punto de entrada de todos los clientes, mejorando el acoplamiento, seguridad, comunicación entre microservicios. La Api Gateway puede llegar a exponer una API para cada cliente diferente para adaptarse mejor a sus requisitos (Ma, Fan, et al., 2018).

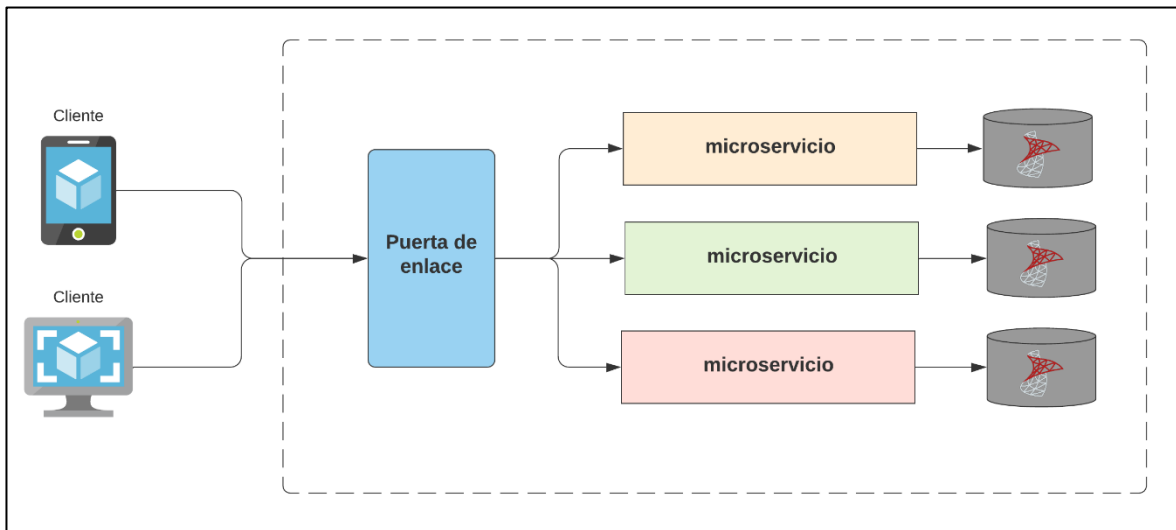


Figura 45. Api Gateway. Fuente: Elaboración propia.

Test Driven Development / Desarrollo guiado por pruebas. – Con TDD, las pruebas se escriben verificando que fallen, se implementa el código haciendo que las pruebas pases satisfactoriamente y luego se refactoriza el código mejorando la estructura y la lógica utilizada. El propósito de TDD es lograr un código que funcione, sea fácil de entender y limpio (Koskela, 2007).

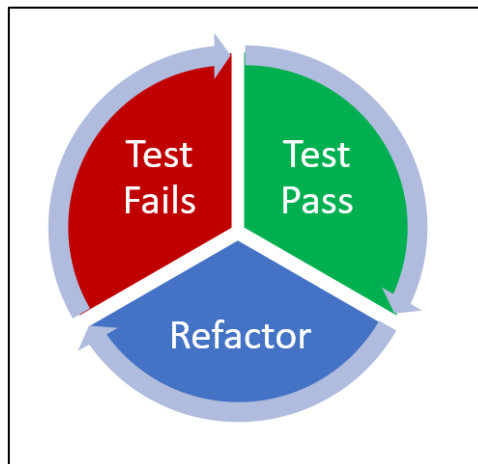


Figura 46. Ciclo TDD. Fuente: Elaboración propia.

Para el correcto desarrollo de los test a continuación se presenta una guía de pruebas y un informe de errores.

Guía de pruebas						
Número y Nombre de Historia de usuario:						
Personas involucradas:						
Ambiente de pruebas:			Cronograma:			
Observaciones:						
Casos de Prueba						
ID	Módulo	Descripción	Ruta	Resultado esperado	Resultado de desarrollo	Resultado de testing

Figura 47. Guía de pruebas. Fuente: Elaboración propia.

Informe de Pruebas		
Número y Nombre de Historia de usuario:		
Personas Involucradas:	Fecha:	
Descripción:		
Pruebas	Número	Porcentaje
Casos de pruebas ejecutadas		
Casos de pruebas con éxito		
Casos de pruebas con error		
Casos de pruebas enviadas para la corrección		

Figura 48. Informe de pruebas. Fuente: Elaboración propia.

MCM.D – Despliegue de Software

Esta actividad comprende el despliegue del software en el cloud, por ello se debe:

- Realizar la revisión del software.
- Adquirir el servicio cloud.
- Realizar el despliegue cloud.

Tabla 15.

Tareas de despliegue cloud.

Roles	Tareas	Entrada	Salida
(LP)	Revisar la historia de usuario,	Historia de usuario	Historia de usuario
(C)	los casos de prueba y el software desarrollado.	Guía de pruebas	[Completa]
		Informe de Pruebas	Guía de pruebas [Revisado]

		Software	Informe de Pruebas [Revisado]
			Software [Revisado]
(LP) (C)	Realizar el análisis para la adquisición del servicio cloud	Ficha de Servicio cloud	Ficha de Servicio cloud [Realizada]
(LP)	Realizar el despliegue del software	Software	Software [Desplegado en el cloud]

Fuente: Elaboración propia.

Los modelos de referencia proporcionan una perspectiva de la tecnología y sus datos, es utilizado para la construcción de modelos similares, proporcionando grados de elementos y disciplinas respectivas (Benedict et al., 2019).

Para el desarrollo de los microservicios se debe de tener un modelo de referencia el cual debe de contar con los siguientes componentes:

Load Balancer / Balanceador de carga. – Permite escalar los microservicios equilibrando la carga y permitiendo que los microservicios estén disponibles, logrando así balancear las solicitudes a las distintas instancias que se generaron (Stubbs, Moreira, & Dooley, 2015).

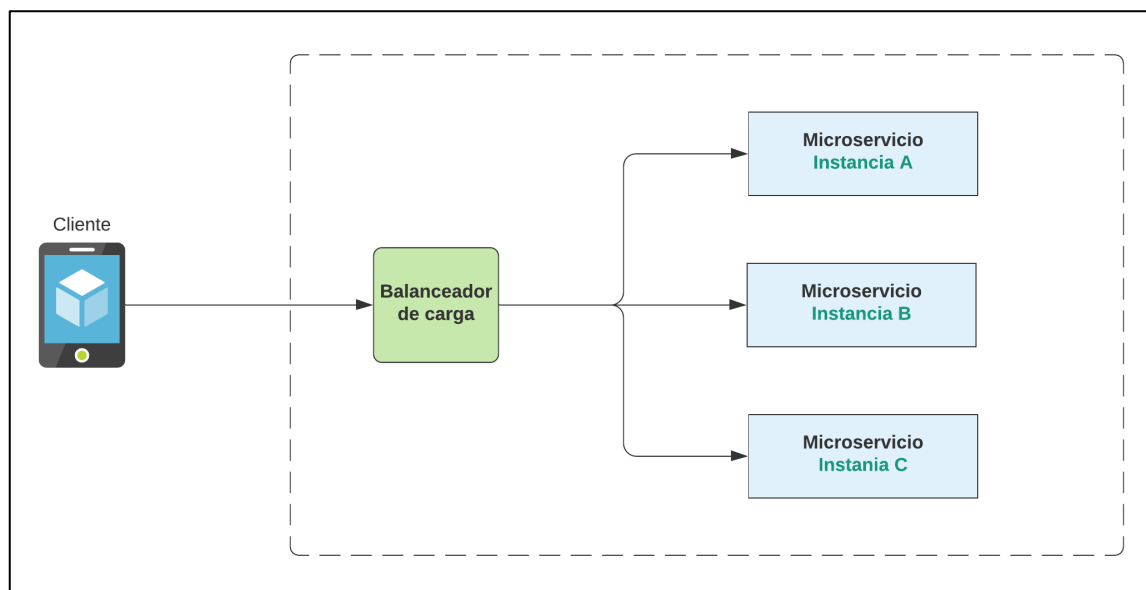


Figura 49. Load Balancer. Fuente: Elaboración propia.

Continuous Integration (CI)/ Integración continua. – Permite preparar y entregar una reléase de manera rápida, ágil y sencilla. La CI requiere que los test realizados sean relevantes para denotar a la calidad y finalidad del código, existen herramientas que permiten la integración continua de manera automatizada (C. Singh et al., 2019).

Continuous Deployment (CD)/ Despliegue continuo. – Permite prealizar un despliegue del software de manera automatizada, para ello se deben de ejecutar una serie de pasos y concluir de manera satisfactoria. (C. Singh et al., 2019).

Servicio Cloud

Los servicios cloud son recursos de tecnologías de la información que funcionan bajo demanda, obteniendo servidores, bases de datos y almacenamiento en una arquitectura soportada por un proveedor de servicios. El uso del cloud cuenta con una amplia gama de usos para diferentes organizaciones para ofrecer distintos servicios. El cloud ofrece agilidad, elasticidad, ahorro en costes de infraestructura, implementación a nivel mundial (AWS, 2019).

Para poder realizar un contrato con un servicio cloud para alojar nuestro software basado en microservicios se deben de tener en cuenta ciertos servicios, por lo que a través de la siguiente ficha se puede realizar la evaluación.

Ficha de Servicio Cloud		
Nombre del servicio:		
Tipo de Servicio:	Costo de servicio	
Descripción		
¿Cuenta con servicios ?	SI	NO
Load Balancer		
Container		
Orchestrator		
Continuous Deployment		

Figura 50. Ficha de Servicio Cloud. Fuente: Elaboración propia.

Desarrollo de la propuesta metodológica.

La validación y desarrollo de la propuesta metodológica se estableció mediante la aplicación de la metodología en un caso de estudio (Moody, 2003). La primera versión de la metodología MCM propuesta se realizó en la empresa HTEC E.I.R.L. La aplicación práctica del caso de estudio se realizó para la aplicación de escritorio GMO, exactamente el módulo de administración de “Control de Ingreso y Salida”, “Persona Observado”, “Punto de entrada” y “Tipo de punto de Entrada” la cual se encuentra desarrollada de manera monolítica en JAVA 8. El análisis se realizó con el equipo de desarrollo de la empresa, dirigida por el investigador principal del proyecto Jersson German Mendoza Linares. A continuación, se muestran los resultados.

A. Análisis de factibilidad de migración de arquitectura de software

El análisis de factibilidad se desarrolló mediante un cuestionario, la reunión se realizó con el jefe de proyectos de tecnologías de información, obteniendo el siguiente resultado.

Ficha de evaluación para la migración de arquitectura		
N°	Pregunta	Respuesta
1	¿De qué manera la arquitectura orientada a microservicios sigue el lineamiento de los objetivos de la empresa?	La arquitectura orientada a microservicios sigue el lineamiento de alta disponibilidad, escalabilidad y mejora continua que estamos dispuestos a brindar a nuestros clientes.
2	¿Cuáles son los objetivos a conseguir?	Mejoras en los procesos, flexibilidad en nuevas integraciones, alta disponibilidad, escalabilidad y mejoras continuas.
3	¿Cómo ayudará la arquitectura orientada a microservicios a alcanzar los objetivos de la empresa?	Brindando una nueva perspectiva en el proceso de desarrollo de software, ofreciendo mejoras continuas a nuestros clientes.
4	¿Cuáles serían las debilidades para realizar la migración?	<ul style="list-style-type: none"> - El desconocimiento de la arquitectura. - Personal no capacitado. - No contar con conocimientos de las tecnologías a utilizar. - Tiempos de desarrollo excesivos.
5	¿De qué manera la arquitectura orientada a microservicios beneficiaría el desarrollo de software constante?	Beneficiaria su escalabilidad, flexibilidad, alta disponibilidad y mejoras continuas.
6	¿Se tienen los recursos económicos adecuados?	Se tiene una solvencia económica sustentable para realizar la migración paulatinamente.
7	¿Cuál es el periodo de tiempo previsto?	Se espera que el tiempo previsto será 6 mes migrando solo algunos módulos.

Figura 51. Ficha de evaluación para la migración de la arquitectura realizada a HTEC E.I.R.L. Fuente: Elaboración propia.

Las preguntas realizadas para evaluar la migración de la arquitectura nos brindaron el contexto para realizar una valorización de la arquitectura actual mediante tres estados, bajo, medio, alto, con la finalidad de valorar la fiabilidad de la migración, si la fiabilidad es alta significa que no es necesario

de realizar una migración de arquitectura de software por lo que la arquitectura actual soporta las necesidades del cliente y del negocio. Sin embargo, si la fiabilidad es baja significa que la arquitectura de software actual no se está aportando satisfactoriamente las necesidades del cliente y del negocio, por lo que tampoco aportará positivamente al negocio, lo cual se traduce en la necesidad de cambiar de arquitectura de software.

Ficha de valoración de la arquitectura actual				
N°	Criterio de valoración de la arquitectura de software actual	Bajo	Medio	Alto
1	Sigue el lineamiento de los objetivos de la empresa	x		
2	Brinda flexibilidad	x		
3	Brinda alta disponibilidad	x		
4	Permite hacer mejoras continuas eficientemente	x		
5	El rendimiento es el esperado		x	
6	Es fácilmente escalable	x		
7	Es versátil cuando se necesita hacer uso de nuevas tecnologías	x		
Total		6	1	0

Figura 52. Ficha de valoración de la arquitectura actual realizada a HTEC E.I.R.L. Fuente: Elaboración propia.

Luego de haber realizado los criterios de valoración de la fiabilidad de la arquitectura actual habiendo obtenido 6 puntos bajos, un punto medio y 0 puntos altos se llegó a la conclusión que actualmente hay un déficit en el software GMO y los lineamientos de la empresa, por ello habiéndose brindado la propuesta de migración de arquitectura a microservicios a través de la MCM propuesta, se acordó que se aplicará la MCM para la migración de GMO a su nueva versión con nombre clave GMO++.

B. Planeamiento de software

Para el planeamiento de software luego de haber aplicado el análisis de factibilidad de migración de arquitectura de software y teniendo los

resultados correspondientes se llevará a cabo el planeamiento, para lo cual nos basaremos en el modelo del Acta de Constitución de la guía de PMBOK, lo cual es un proceso para definir la existencia de un proyecto, sirviendo como directriz para tener una concepción clara, sencilla y formal de lo que se va a realizar, además brindar información vital y dar la autoridad para la asignación de los recursos necesarios para llevar a cabo las actividades.

Hoja de Planificación			
Nombre del proyecto	Migración de arquitectura a microservicios	Fecha	01/09/2021
Responsable	Jersson German Mendoza Linares	Sponsor	HTEC E.I.R.L.
Financiamiento/ Coste	S/.1500		
Descripción			
Se realizará una migración de arquitectura, actualmente el software se encuentra desarrollado de una manera monolítica lo cual implica que se encuentra en un solo núcleo de código, por ello se realizará la migración a la arquitectura a microservicios para solventar las debilidades que se presentan actualmente. Solo se realizará la migración del Módulo de “Control de Ingresos y Salida”, el cual contiene la administración de “Personal Observado”, “Punto de Entrada”, “Tipo de Punto de Entrada”, esto como prueba de la metodología MCM y su posible adaptación en el futuro.			
Justificación			
El software actual no cumple con las expectativas ni sigue el lineamiento del negocio, por ello se realizará la migración a microservicios.			
Roles			
N°	Nombre Completo	Rol	
1	Kevin Leandro Vilcherrez Chavarry	Cliente	
2	Kevin Leandro Vilcherrez Chavarry	Experta de dominio	
3	Jersson German Mendoza Linares	Líder de programación	
4	Equipo de desarrollo HTEC	Programador	
5	Jersson German Mendoza Linares	Encargado de pruebas	
Riesgos y Limitaciones			
El costo del proyecto de migración, tiempo de desarrollo sobreestimados y el personal sin conocimientos básicos de las tecnologías a utilizar.			

Figura 53. Hoja de planificación realizada a HTEC E.I.R.L. Fuente: Elaboración propia.

Después de haber llenado el documento de acta de constitución se realizó el proceso de levantamiento de requerimientos. Para dichas actividades se hizo uso de las historias de usuario, esto permitió describir en un lenguaje

común los requisitos, características, funcionalidades del proceso a desarrollar.

Historia de Usuario			N° H0001
Nombre de historia: CIS – Administración de Punto de Entrada			
Usuario: Personal de recursos humanos			
Prioridad de desarrollo: Alto	Riesgo de desarrollo: Bajo	Dependencia: --	
Tiempo estimado: 2d		Número de Iteración: 1	
Responsable: Jerisson German Mendoza Linares			
Descripción			
Listar Punto de entradas			
Se requiere desarrollar un formulario donde se pueda visualizar el listado de puntos de entradas. (Todos los listados tienen filtros por cada columna en las tablas y adicional un buscador).			
Campo Filtro	Columna BD	Filtro	Descripción
c1	activo	Activo	lista desplegable (activo, inactivo, todos)
<div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> TODOS Buscar <input style="width: 100px;" type="text"/> </div>			
Luego de seleccionar los filtros se mostrará la información de los puntos de entradas con los siguientes campos:			
Tabla	Campo	Nombre	Descripción
puntoentrada	idpuntoentrada	ID PUNTO ENTRADA	Identificador del punto entrada (No se muestra)
puntoentrada	codigo	CÓDIGO	código de punto de entrada
puntoentrada	descripcion	DESCRIPCIÓN	descripción del punto de entrada
tipopuntoentrada	idtipopuntoentrada	ID TIPO PUNTO DE ENTRADA	Identificador del tipo de punto entrada (No se muestra)
tipopuntoentrada	descripcion	DESCRIPCIÓN TIPO PUNTO ENTRADA	descripción del tipo de punto de entrada

puntoentrada	activo	ACTIVO	Estado del punto de entrada
--------------	--------	--------	-----------------------------

Crear Punto de Entrada

Validaciones (Check):

Campo Filtro	Columna BD	Descripción
c1	código	Verificar la existencia de un punto de entrada con el código ingresado por empresa

Se requiere desarrollar un formulario donde se ingrese los siguientes campos

Columna BD	Tipo	Descripción	Vista	Validación
idempresa	varchar(3)	Identificador de la empresa con la que se logueo. (Se obtiene del TOKEN)	NO	Formato numérico, sin tildes, sin caracteres especiales ni espacios, ni guiones, 3 caracteres
idusuario	int	Identificador del usuario con el que inició sesión. (Se obtiene del TOKEN)	NO	Valor autoincremental
idpuntoentrada	varchar(36)	Identificador del punto de entrada (campo por defecto desde bd)	NO	Formato alfanumérico, 36 caracteres
codigo	varchar(10)	código de punto de entrada	input	Formato alfanumérico, mínimo 3 caracteres y máximo 10 caracteres obligatorio. Debe ser ÚNICO por empresa Campo Obligatorio
descripcion	varchar(50)	nombre o descripción del punto de entrada	input	formato libre de validación textual, mínimo 3 caracteres y máximo 50 caracteres Campo Obligatorio
idtipopuntoentrada	varchar(36)	Identificador del tipo punto de entrada	select	Campo Obligatorio
activo	bit	campo por defecto desde bd (1)	NO	0 inactivo 1 activo
fechacreacion	datetime2(0) not null	campo por defecto desde bd(getdate())	NO	Formato YYYY-mm-dd hh:mm:ss, 18 caracteres

Nuevo Punto Entrada

Ingresar IDPTOENTRADA

minimo 0 / 10

Ingresar DESCRIPCIÓN

minimo 0 / 50

Seleccionar Tipo Punto Entrada



Validación: el código deberá ser único por empresa

Editar Punto de Entrada

Se requiere desarrollar un formulario donde se ingrese la descripción, y seleccionar el tipo de punto de entrada.

Columna BD	Tipo	Descripción	Vista	Validación
idempresa	varchar(3)	Identificador de la empresa con la que se logueo. (Se obtiene del TOKEN)	NO	Formato numérico, sin tildes, sin caracteres especiales ni espacios, ni guiones, 3 caracteres
idusuario	int	Identificador del usuario con el que inició sesión. (Se obtiene del TOKEN)	NO	Valor autoincremental
idpuntoentrada	varchar(36)	Identificador del punto de entrada (Se envía desde frontend)	NO	Formato alfanumérico, 36 caracteres
codigo	varchar(10)	código de punto de entrada	input no editable	Formato alfanumérico, mínimo 3 caracteres y máximo 10 caracteres obligatorio. Debe ser ÚNICO por empresa
descripcion	varchar(50)	nombre o descripción del punto de entrada	input	formato libre de validación textual, mínimo 3 caracteres y máximo 50 caracteres Campo Obligatorio
idtipopuntoentrada	varchar(36)	Identificador del tipo punto de entrada	select	Campo Obligatorio
activo	bit	campo por defecto desde bd (1)	NO	0 inactivo 1 activo

fechaalteracion	datetime2(0)	campo fecha de modificación en la base de datos (campo por defecto desde bd)	NO	Formato YYYY-mm-dd hh:mm:ss, 18 caracteres
-----------------	--------------	--	----	--

Editar Punto Entrada

Ingresar IDPTOENTRADA

001

3 / 10

Ingresar DESCRIPCIÓN

Garita 1

8 / 50

Entrada Normal



Validación: el código no podrá ser editado

Anulación Punto de Entrada

Se requiere desarrollar un formulario donde se visualice un mensaje hacia el usuario donde especifique si se encuentra seguro o no: ¿Estás seguro de anular el punto de entrada?

Activar Punto de Entrada

Se requiere desarrollar un formulario donde se visualice un mensaje hacia el usuario donde especifique si se encuentra seguro o no: ¿Estás seguro de activar el punto de entrada?

Activar y anular punto de entrada:

Campo	Tipo	Descripción	Vista	Validación
idempresa	varchar(3)	Identificador de la empresa con la que inició sesión. (Se obtiene del TOKEN)	NO	Formato alfanumérico, sin tildes, sin caracteres especiales ni espacios, ni guiones, 3 caracteres
idusuario	int	Identificador del usuario con el que inició sesión. (Se obtiene del TOKEN)	NO	Valor autoincremental
idpuntoentrada	varchar(36)	Identificador del tipo de punto de entrada (Se envía desde frontend)	data del listado	Formato alfanumérico, 36 caracteres
activo	bit	Valor al que se pasará el valor activo (Se envía desde frontend)	data del listado	true o false. Envía el valor al estado que este se requiera si es activo equivale a true e inactivo a false.

Opciones tabla puntos de entrada

Se requiere hacer una opción usando el click derecho al seleccionar un registro y deberá listar las siguientes opciones.

Opción	Descripción
Editar	al hacer click se abrirá el modal de editar punto de entrada
Anular	al hacer click se abrirá el modal de diálogo de confirmación de la anulación
Activar	Al hacer click derecho se abrirá el modal de diálogo de confirmación de la activación

Teclas especiales

Crear Atajos de teclado

Tecla	Función
"F2"	Abrirá el crear punto de entrada
"F6"	Abrirá el editar punto de entrada
"F7"	Abrirá el modal de anulación

Validación

Feature: Gestionar Punto de entrada

Scenario: insertar y editar punto de entrada
Glven que estoy en la interfaz
When los inputs se encuentran vacíos o no cumplen con las reglas de validaciones de longitud y formato
And se solicita el resultado al cliente
Then en los campos del formulario debería aparecer el mensaje de campo requerido o el mensaje de restricción por longitud o formato en todos los campos del formulario en la vista respectivamente

Glven que estoy en la interfaz
When los campos requeridos del formulario no están vacíos
And si las validaciones en el cliente están correctas se solicita el resultado al servidor
Then muestra el mensaje de respuesta que le devuelve api

Scenario: eliminar punto de entrada
Glven que estoy en la interfaz
When hago un click sobre el icono de eliminar motivo se mostrará un mensaje de confirmación
And se confirma el mensaje se solicita el resultado al servidor
Then muestra el mensaje de respuesta que le devuelve api

Estado: 0%

Figura 54. Historia de usuario de Administración de Punto de Entrada.
Fuente: Elaboración propia.

Nombre de historia: CIS – Administración de Tipos de Punto de Entrada

Usuario: Personal de recursos humanos

Prioridad de desarrollo: Alto

Riesgo de desarrollo: Bajo

Dependencia: --

Tiempo estimado: 2d

Número de Iteración: 1

Responsable: Jerisson German Mendoza Linares

Descripción

Listar Tipos Punto Entrada

Se requiere desarrollar un formulario donde se pueda visualizar el listado de tipos de puntos de entradas. (Todos los listados tienen filtros por cada columna en las tablas y adicional un buscador).

Campo Filtro	Columna BD	Filtro	Descripción
c1	activo	Activo	lista desplegable (activo, inactivo, todos)



TODOS



Buscar



Luego de seleccionar los filtros se mostrará la información de los tipos de puntos de entradas con los siguientes campos:

Tabla	Columna BD	Campo	Descripción
tipopuntoentrada	idtipopuntoentrada	ID TIPO PUNTO ENTRADA	Identificador del tipo punto entrada (No se muestra)
tipopuntoentrada	codigo	CODIGO	código del tipo de punto de entrada
tipopuntoentrada	descripcion	DESCRIPCION	Descripción del tipo de punto de entrada
tipopuntoentrada	esenviaalerta	ES ENVIAR ALERTA	si envía alertas a través de los métodos de las apps como cis
tipopuntoentrada	esgarita	ES GARITA	es obligatorio para garita
tipopuntoentrada	esbus	ES BUS	si es para buses
tipopuntoentrada	espretareador	PRETAREADO	si es pretareador
tipopuntoentrada	esObligaAsistencia	ES OBLIGATORIO ASISTENCIA	si es obligatorio asistencia
tipopuntoentrada	activo	ACTIVO	Estado del tipo de punto de entrada

Crear Tipo Punto de Entrada

Validaciones (Check):

Campo Filtro	Columna BD	Descripción
c1	código	Verificar la existencia de un tipo de punto de entrada con el código ingresado

Se requiere desarrollar un formulario donde se ingrese los siguientes campos

Columna BD	Tipo	Descripción	Vista	Validación
idempresa	varchar(3)	Identificador de la empresa con la que se logueo. (Se obtiene del TOKEN)	NO	Formato numérico, sin tildes, sin caracteres especiales ni espacios, ni guiones, 3 caracteres
idtipopuntoentrada	varchar(36)	Identificador del tipo de punto de entrada (campo por defecto desde bd)	NO	Formato alfanumérico, 36 caracteres
codigo	varchar(3)	código del tipo de punto de entrada	input	formato alfanumérico, mínimo 3 caracteres y máximo 3 caracteres. Debe ser ÚNICO por empresa Campo Obligatorio
descripcion	varchar(25)	nombre o descripción del tipo de punto de entrada	input	formato libre de validación textual, mínimo 3 caracteres y máximo 25 caracteres Campo Obligatorio
esenviaalerta	smallint	campo envía alertas del tipo de punto de entrada	checkbox	numérico, 1 - seleccionado 0 no seleccionado
esgarita	smallint	campo garita del tipo de punto de entrada	checkbox	numérico, 1 - seleccionado 0 no seleccionado

espretareador	smallint	campo pretareador del tipo de punto de entrada	checkbox	numérico, 1 - seleccionado 0 no seleccionado
esbus	smallint	campo bus del tipo de punto de entrada	checkbox	numérico, 1 - seleccionado 0 no seleccionado
esobligatorioasistencia	smallint	campo obligatorio del tipo de punto de entrada	checkbox	numérico, 1 - seleccionado 0 no seleccionado
activo	bit	campo por defecto desde bd (1)	NO	Formato numérico, 0 inactivo 1 activo
fechacreacion	datetime2(0) not null	campo fecha por default al insertar en la bd(GETDATE())	NO	Formato YYYY-mm-dd hh:mm:ss, 18 caracteres

Validación: el código será único, los otros demás datos serán por sus valores por default.

Editar Punto de Entrada

Validaciones (Check):

Campo Filtro	Columna BD	Descripción
c1	código	Verificar la existencia de un tipo de punto de entrada con el código ingresado.

Se requiere desarrollar un formulario donde se puedan editar los siguientes campos.

Columna BD	Tipo	Descripción	Vista	Validación
idempresa	varchar(3)	Identificador de la empresa con la que se logueo. (Se obtiene del TOKEN)	NO	Formato numérico, sin tildes, sin caracteres especiales ni espacios, ni guiones, 3 caracteres
idtipopuntoentrada	varchar(36)	Identificador del tipo de punto de entrada (Se envía desde frontend)	NO	Formato alfanumérico, 36 caracteres
codigo	char(3)	código del tipo de punto de entrada	input - NO EDITABLE	formato alfanumérico, mínimo 3 caracteres y máximo 3 caracteres. CAMPO NO EDITABLE

descripcion	varchar(25)	nombre o descripción del tipo de punto de entrada	input	ormato libre de validación textual, mínimo 3 caracteres y máximo 25 caracteres Campo Obligatorio
esenviaalerta	smallint	campo envía alertas del tipo de punto de entrada	checkbox	numérico, 1 - seleccionado 0 no seleccionado
esgarita	smallint	campo garita del tipo de punto de entrada	checkbox	numérico, 1 - seleccionado 0 no seleccionado
espretareador	smallint	campo pretareador del tipo de punto de entrada	checkbox	numérico, 1 - seleccionado 0 no seleccionado
esbus	smallint	campo bus del tipo de punto de entrada	checkbox	numérico, 1 - seleccionado 0 no seleccionado
esobligatorioasistencia	smallint	campo obligatorio del tipo de punto de entrada	checkbox	numérico, 1 - seleccionado 0 no seleccionado
fechaalteracion	datetime2(0)	campo fecha por default en la bd (getedate())	NO	Formato YYYY-mm-dd hh:mm:ss, 18 caracteres

Validación: el código no podrá ser editado

Anulación Tipo Punto de Entrada

Se requiere desarrollar un formulario donde se visualice un mensaje hacia el usuario donde especifique si se encuentra seguro o no: ¿Estás seguro de eliminar Tipo de punto de entrada?

Activar Tipo Punto de Entrada

Se requiere desarrollar un formulario donde se visualice un mensaje hacia el usuario donde especifique si se encuentra seguro o no: ¿Estás seguro de activar el Tipo de punto de entrada?

Activación y anulación de tipo de punto de entrada:

Campo	Tipo	Descripción	Vista	Validación
idempresa	varchar(3)	Identificador de la empresa con la que inició sesión. (Se obtiene del TOKEN)	NO	Formato alfanumérico, sin tildes, sin caracteres especiales ni espacios, ni guiones, 3 caracteres
idusuario	int	Identificador del usuario con el que inició sesión. (Se obtiene del TOKEN)	NO	Valor autoincremental
idtipopuntoentrada	varchar(36)	Identificador del tipo	data	Formato alfanumérico,

		de punto de entrada (Se envía desde frontend)	del listado	36 caracteres
activo	bit	Valor al que se pasará el valor activo (Se envía desde frontend)	data del listado	true o false. Enviar el valor true si el registro está desactivado y false si el registro está activo.

Opciones tabla tipos puntos de entrada

Se requiere hacer una opción usando el click derecho al seleccionar un registro y deberá listar las siguientes opciones.

Opción	Descripción
Editar	al hacer click se abrirá el modal de editar punto de entrada
Anular	al hacer click se abrirá el modal de diálogo de confirmación de la anulación
Activar	Al hacer click derecho se abrirá el modal de diálogo de confirmación de la activación

Teclas especiales

Crear Atajos de teclado

Tecla	Función
"F2"	Abrirá el crear tipo punto de entrada
"F6"	Abrirá el editor tipo punto de entrada
"F7"	Abrirá el modal de anulación

Validación

Feature: Gestión de Tipo punto de Entrada

Scenario: Listar tipo punto de entrada

Glven que estoy en la interfaz

When seleccionas el estado y no tiene registros con el estado indicado

And se solicita los registros

Then no se mostrará información y/o algún mensaje de error

Glven que estoy en la interfaz

When seleccionas el estado correcto

And se solicita los registros

Then muestra todos los registros del tipo punto de entrada

Scenario: insertar tipo punto de entrada

Glven que estoy en la interfaz

When los inputs se encuentran vacíos o no cumplen con las reglas de validaciones de longitud y formato

And se solicita el resultado al cliente

Then en los campos del formulario debería aparecer el mensaje de campo requerido o el mensaje de restricción por longitud o formato en todos los campos del formulario en la vista respectivamente

<p>Glven que estoy en la interfaz When los campos requeridos del formulario no están vacíos And si las validaciones en el cliente están correctas se solicita el resultado al servidor Then muestra el mensaje de respuesta que le devuelve api</p> <p>Scenario: activar y desactivar tipo punto de entrada Glven que estoy en la interfaz When hago un click sobre el icono de activar o desactivar el tipo punto de entrada se mostrará un mensaje de confirmación And se confirma el mensaje se solicita el resultado al servidor Then muestra el mensaje de respuesta que le devuelve api</p>
<p>Estado: 0%</p>

Figura 55. Historia de usuario de Administración de Tipos de Punto de Entrada. Fuente: Elaboración propia.

Historia de Usuario		N° H0003	
Nombre de historia: CIS – Administración de Personal Observado y BL			
Usuario: Personal de recursos humanos			
Prioridad de desarrollo: Alto	Riesgo de desarrollo: Bajo	Dependencia: --	
Tiempo estimado: 3d		Número de iteración: 1	
Responsable: Jersson German Mendoza Linares			
Descripción			
Listar Personal Observado			
Se requiere desarrollar un formulario donde se pueda visualizar el listado de personal observado. (Todos los listados tienen filtros por cada columna en las tablas y adicional un buscador).			
Campo Filtro	Columna BD	Filtro	Descripción
c1	fechainicio	Fecha Inicio	campo de rango fecha por default (día actual del sistema).
c2	fechainicio	Fecha Fin	campo de rango fecha por default (día actual del sistema).
c3	idplanilla	Planillas	Lista desplegable múltiple con información de las planillas
c4	activo	Activo	Lista desplegable, contiene los valores (Todos - vacío, Activos - true, Inactivos - false)
Luego de seleccionar el filtro se mostrará la información del personal observado con los siguientes campos:			
Tabla	Columna BD	Campo	Descripción
trabajadorobservadoxplanilla	idtrabajadorobservadoxplanilla	ID TRABAJADOR OBSERVADO PLANILLA	Identificador del trabajador observado planilla (<u>No se muestra</u>)
trabajadorobservadoxplanilla	idplanilla	ID PLANILLA	Información de Planilla
tipomotivox	descripcion	TIPO MOTIVO	descripción del tipo de motivo

trabajadorobservado	numerodocumento	NUMERO DOCUMENTO TRABAJADOR	número de documento del trabajador (DNI, CARNET EXTRANJERÍA, OTROS)
trabajador	nombresall	NOMBRE COMPLETO	Nombres del trabajador
trabajadorobservado	fechainiciosuspension	FECHA INICIO	Fecha inicio del personal observado
trabajadorobservado	fechainiciosuspension	FECHA FIN	Fecha fin del personal observado
trabajadorobservado	motivo	MOTIVO	Descripción del Motivo de la observación del personal
trabajadorobservado	descripmotivo	DETALLE OBSERVACION	Descripción del motivo de la observación del personal
trabajadorobservado	observaciones	OBSERVACIONES	Observaciones adicionales del personal
trabajadorobservadoxplanilla	activo	ACTIVO	Estado del trabajador Observado

Añadir Personal Observado y Personal de Black List

- Búsqueda (Get):

Campo Filtro	Columna de BD	Descripción
-	idempresa	Código de la empresa (se obtiene del token)
c1	dnitabajadores	un array de los trabajadores conteniendo el número de documento de cada trabajador

Esta búsqueda es para encontrar los nombres de los trabajadores que se encuentren registrados, (si el trabajador no se encuentra registrado devolver "Trabajador no identificado").

- Validación en la búsqueda (Check):
Se efectuarán las siguientes validaciones con el número de documento de los trabajadores digitando o importando.
 - Formato correcto del número de documento (formato numérico).
 - Cantidad de dígitos de dni mayor a 7 y menor a 13 dígitos.

Se requiere desarrollar un formulario donde se ingrese la planilla, fecha inicio, fecha fin, motivo de la observación, si es alerta, trabajadores, detalle de la observación y comentarios.

Columna BD	Tipo	Descripción	vista	Validación
idempresa	varchar(3)	Identificador de la empresa con la que se logueo. (Se obtiene del <u>TOKEN</u>)	NO	Formato alfanumérico , sin tildes, sin caracteres especiales ni espacios, ni guiones, 3 caracteres
idtrabajadorobservado	varchar(36)	Identificador del trabajador observado (Se obtiene mediante bd)	NO	Formato alfanumérico , 36 caracteres
idtrabajadorobservadoplanilla	varchar(36)	Identificador del trabajador observadoxplanilla (Se obtiene mediante bd)	NO	Formato alfanumérico , 36 caracteres
idplanilla	varchar(10)	Identificador de la planilla	Planilla - Multiple Select	Formato alfanumérico , 10 caracteres
idtipomotivo	int	Identificador del tipo motivo	Tipo Motivo - select	Formato numérico
idusuario	varchar(25)	Identificador del usuario con la que se logueo. (Se obtiene del <u>TOKEN</u>)	NO	Formato alfanumérico , 25 caracteres
idtrabajador	varchar(25)	Identificador del trabajador (Se obtiene mediante bd)	NO	Formato alfanumérico , 25 caracteres

numerodocumento	varchar(12)	número del documento de identidad	Listado Trabajadores - Table	formato dni numérico 8 caracteres, debe ser ÚNICO por empresa, alfanumérico carnet extranjeria 12 caracteres
<i>Se tendrá la opción de digitar un número de documento que lo añadirá a un listado.</i>				
descripcion	varchar(50)	campo por defecto desde bd (<u>SUSPENDIDO EN GMO</u>)	NO	Formato alfanumérico , 50 caracteres
nombres	varchar(300)	campo por defecto desde bd (<u>NOMBRE DEL TRABAJADOR</u>)	NO	Formato alfanumérico , 300 caracteres
fechainicio	DATE	Fecha inicio de registro de la observación	input	formato YYYY-mm-d, 8 caracteres
suspension	varchar(2)	campo por defecto desde bd (<u>SI</u>)	NO	Formato alfabético, 2 caracteres
motivo	varchar(300)	campo por defecto desde bd (<u>descripción del motivo por el idmotivo</u>)	NO	seleccionar el id del motivo (vista, no editable)
descripcionmotivo	varchar(400)	detalle de la observación del trabajador observado	input	Formato alfanumérico , 400 caracteres
observaciones	nvarchar(max)	observación del trabajador observado	input	Formato alfanumérico , max caracteres
fechainiciosuspension	date	Fecha inicio suspensión del registro de la observación (campo por defecto fechainicio)	NO	formato YYYY-mm-d, 8 caracteres

fechafinsuspension	date	fecha fin de la observación del trabajador observado	input	formato YYYY-mm-d, 8 caracteres
package	varchar(25)	campo por defecto desde bd (GMO)	NO	formato alfabético, 25 caracteres.
activo	bit	campo por defecto desde bd (1)	NO	Formato numérico, 0 inactivo 1 activo

Agregar Personal Observado

Planilla: OBREROS PACKING

Seleccione Motivo Observación: DERIVAR A GESTIÓN HUMANA

Trabajadores Observados

Ingresar Detalle de la Observación:

Ingresar comentario u observación:

Fecha Inicio: 2021-08-31

Fecha Fin: 2021-08-31

Alerta:

Anulación y activación del Personal Observado

Se requiere desarrollar un formulario donde se visualice un mensaje hacia el usuario donde especifique si se encuentra seguro o no: ¿Estás seguro de eliminar el personal observado?

- La anulación y activación del personal observado es múltiple.

Campo	Tipo	Descripción	Vista	Validación
idempresa	varchar(3)	Identificador de la empresa con la que inició sesión. (Se obtiene del <u>TOKEN</u>)	NO	Formato alfanumérico, sin tildes, sin caracteres especiales, espacios, ni ni

				guiones, caracteres. 3
idusuario	int	Identificador del usuario con el que inició sesión. (Se obtiene del <u>TOKEN</u>)	NO	Valor autoincremental
idtrabajadorobservadoplanilla	varchar(36)	Identificador del trabajador observado por planilla	Data del Listado	Formato alfanumérico, 36 caracteres
activo	bit	Valor al que se pasará el valor activo	Data del Listado	true o false. Enviar el valor true si el registro está desactivado y false si el registro está activo.

Exportación Personal Observado

Se requiere una ventana donde permite exportar la información del personal en un Excel y la ubicación del archivo

Teclas especiales

Crear Atajos de teclado

Tecla	Función
"F2"	Se visualizará una ventana para una búsqueda
"F3"	Abrirá el añadir un personal observado
"D"	Se visualizará el mensaje de anulación del registro

Validación

Feature: Gestión de Personal Observado

Scenario: Listar Personal Observado

Given que estoy en la interfaz

When seleccionas la fecha y no tiene registros con el estado y/o la planilla indicada

And se solicita los registros

Then no se mostrará información y/o algún mensaje de error

Given que estoy en la interfaz

When seleccionas todo las fechas y la planilla correcta

And se solicita los registros

Then muestra todos los registros del personal

Scenario: insertar personal observado

<p> Glven que estoy en la interfaz When los inputs se encuentran vacíos o no cumplen con las reglas de validaciones de longitud y formato And se solicita el resultado al cliente Then en los campos del formulario debería aparecer el mensaje de campo requerido o el mensaje de restricción por longitud o formato en todos los campos del formulario en la vista respectivamente </p> <p> Glven que estoy en la interfaz When los campos requeridos del formulario no están vacíos And si las validaciones en el cliente están correctas se solicita el resultado al servidor Then muestra el mensaje de respuesta que le devuelve api </p> <p> Scenario: activar y desactivar personal observado Glven que estoy en la interfaz When hago un click sobre el icono de activar o desactivar el personal observado se mostrará un mensaje de confirmación And se confirma el mensaje se solicita el resultado al servidor Then muestra el mensaje de respuesta que le devuelve api </p>
<p>Estado: 0%</p>

Figura 56. Historia de usuario de Administración de Personal Observado y BL. Fuente: Elaboración propia.

Al haber documentado los requerimientos a través de las historias de usuario se planificará el primer sprint para planificar la duración y al finalizar generar un entregable del producto

Planificación de Sprint				
Sprint	Historia	Nombre de la Historia de Usuario	Nivel de prioridad	Duración estimada
S0001	H0001	CIS – Administración de Punto de Entrada	Alta	2d
	H0002	CIS – Administración de Tipos de Punto de Entrada	Alta	2d
	H0003	CIS – Administración de Personal Observado y BL	Alta	3d
Total				7d

Figura 57. Ficha de planificación de sprint. Fuente: Elaboración propia.

C. Diseño de software

Durante el diseño de software se realizó un cuadro comparativo de las tecnologías actuales y a las que se proponen para la migración para poder brindar un mayor enfoque y contexto, la aplicación monolítica actual se encuentra desarrollada en Java 1.8 de escritorio. Para realizar una propuesta en el lado del Front-end (cliente), se realizó una investigación profunda de la tecnología propuesta la cual es VueJS.

Ficha de comparación de tecnologías		
Tecnología Comparación	Java Actual	Vuejs Propuesta
Definición	Java es una tecnología que se usa comúnmente para el desarrollo de aplicaciones de escritorio. Java no es lo mismo que javascript, que se trata de una tecnología sencilla que se usa para crear páginas web y solamente se ejecuta en el explorador.	Vue es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente.
Uso	Escritorio	Web y Progressive web apps
Plataforma	Windows	Navegador Web, Android, Windows, Mac
Integración	Se integra con otras fuentes de datos internamente en el sistema a través del JDBC	Se integra con otras plataformas a través de la arquitectura REST
Licencia	Licencia condicional según términos y condiciones de Oracle	Open source
Comunidad	Amplia comunidad, documentación y años en el mercado	Amplia comunidad y documentación, se encuentra en las últimas tecnologías del mercado
Peticiones	-----	-----
Tiempo de respuesta	-----	-----

Figura 58. Ficha de comparación de tecnologías en el front-end. Fuente: Elaboración propia.

En el lado del servidor se propuso hacer uso de Lumen, para ello se realizó una investigación profunda de la tecnología.

Ficha de comparación de tecnologías		
Comparación \ Tecnología	Java Actual	Lumen Propuesta
Definición	Java es una tecnología que se usa comúnmente para el desarrollo de aplicaciones de escritorio. Java no es lo mismo que javascript, que se trata de una tecnología sencilla que se usa para crear páginas web y solamente se ejecuta en el explorador.	Lumen es un micro-framework para PHP que comparte muchos de los componentes de su «hermano mayor» Laravel, pero Lumen es una versión más liviana de Laravel y orientado más a la creación de APIs y microservicios,
Uso	Escritorio	Servidor
Plataforma	Windows	Windows, Linux, Mac
Integración	Se integra con otras fuentes de datos internamente en el sistema a través del JDBC	Se integra con fuentes de datos proveniente de bases de datos o incluso de APIS
Licencia	Licencia condicional según términos y condiciones de Oracle	Open Source
Comunidad	Amplia comunidad, documentación y años en el mercado	Amplia comunidad y documentación, se encuentra en las últimas tecnologías del mercado
Peticiones	-----	19000/s
Tiempo de respuesta Describe cuánto es el tiempo de respuesta por segundo	-----	±10.3/ms

Figura 59. Ficha de comparación de tecnologías en el back-end. Fuente: Elaboración propia.

Para conectar el Front-end y Backend se hizo uso de la arquitectura REST, esta arquitectura es flexible, con directrices y soporte para múltiples estándares, como HTTPS, JSON y XML. Sin embargo, para la comunicación se hará uso del formato JavaScript Object Notation (JSON) siendo un formato de datos ligero y mínimo utilizado para almacenar y transportar

datos. JSON se utiliza con mayor frecuencia para enviar datos desde un servidor a una página web o aplicación.

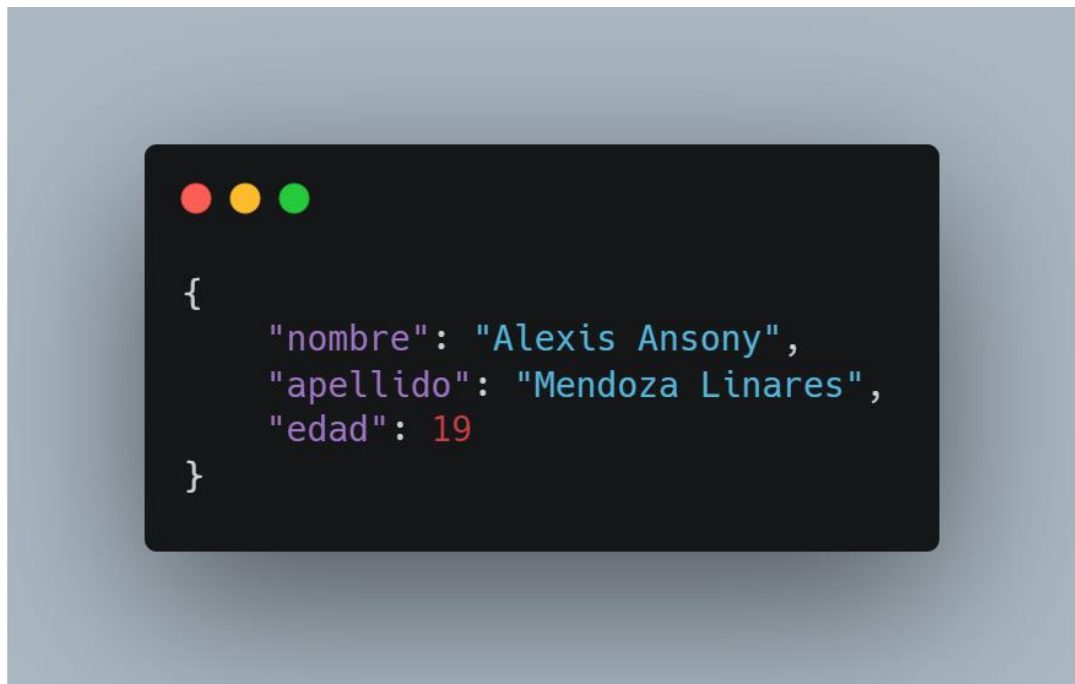


Figura 60. Estructura de formato JSON. Fuente: Elaboración propia.

REST se identifica mediante direcciones globales y únicas o identificadores de identidad a las que se les llama URI. Esta interfaz uniforme necesita saber el estado del recurso que ha sido recuperado, esto puede ser recuperado por medio de los metadatos, para así poder saber cómo se debe analizar, procesar y comprender la representación. Los mensajes están sujetos a un protocolo de red el cual indica como deben procesarse, el protocolo HTTP define un conjunto de métodos y un conjunto de códigos.

Tabla 16.

Métodos HTTP.

Métodos	Descripción
GET	El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.
HEAD	El método HEAD pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.
POST	El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.
PUT	El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.
PATCH	El método PATCH es utilizado para aplicar modificaciones parciales a un recurso.
DELETE	El método DELETE borra un recurso en específico.

Fuente: Elaboración propia

Tabla 17.

Códigos HTTP.

Código	Descripción
1XXX	Código informativo
2XXX	Código de peticiones exitosas
3XXX	Código de redirección
4XXX	Código de error del cliente
5XXX	Código de errores de servidor

Fuente: Elaboración propia.

Los diagramas BPMN permiten tener un contexto del dominio del negocio de manera gráfica, esto permitió entender de manera simple el flujo de cada proceso, se realizaron los siguientes diagramas:

CONTROL DE INGRESO Y SALIDA

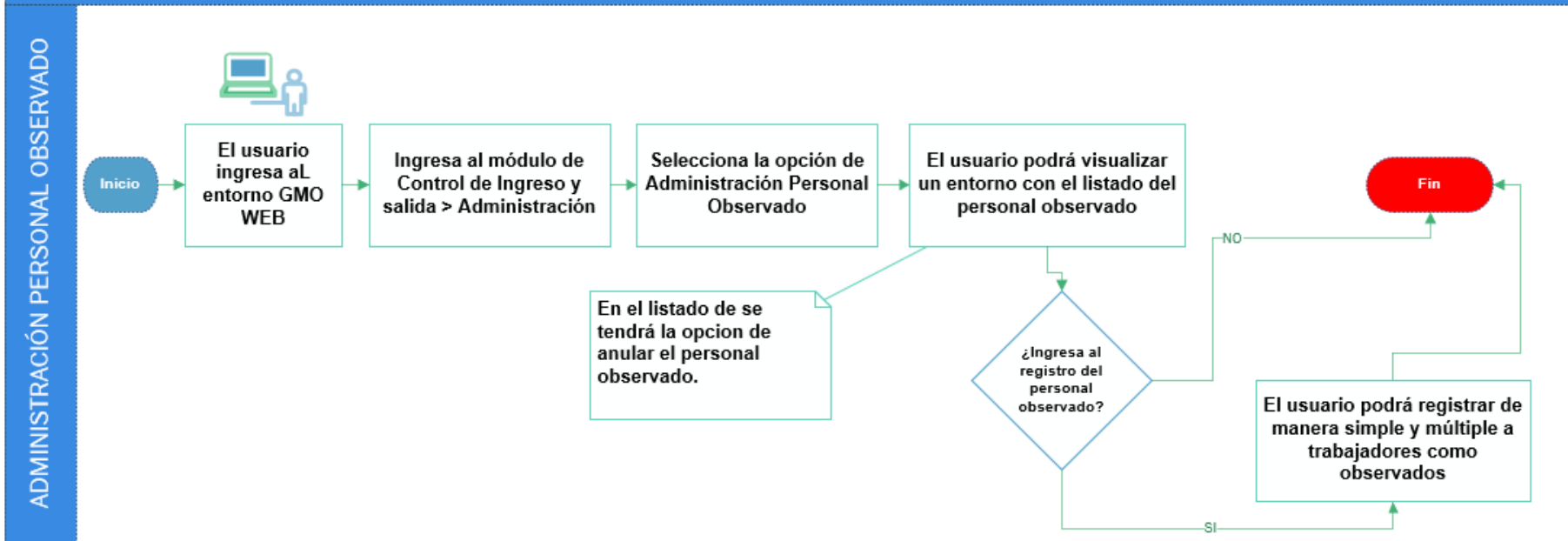


Figura 61. Diagrama BPMN de la Administración de Personal Observado. Fuente: Elaboración propia

CONTROL DE INGRESO Y SALIDA

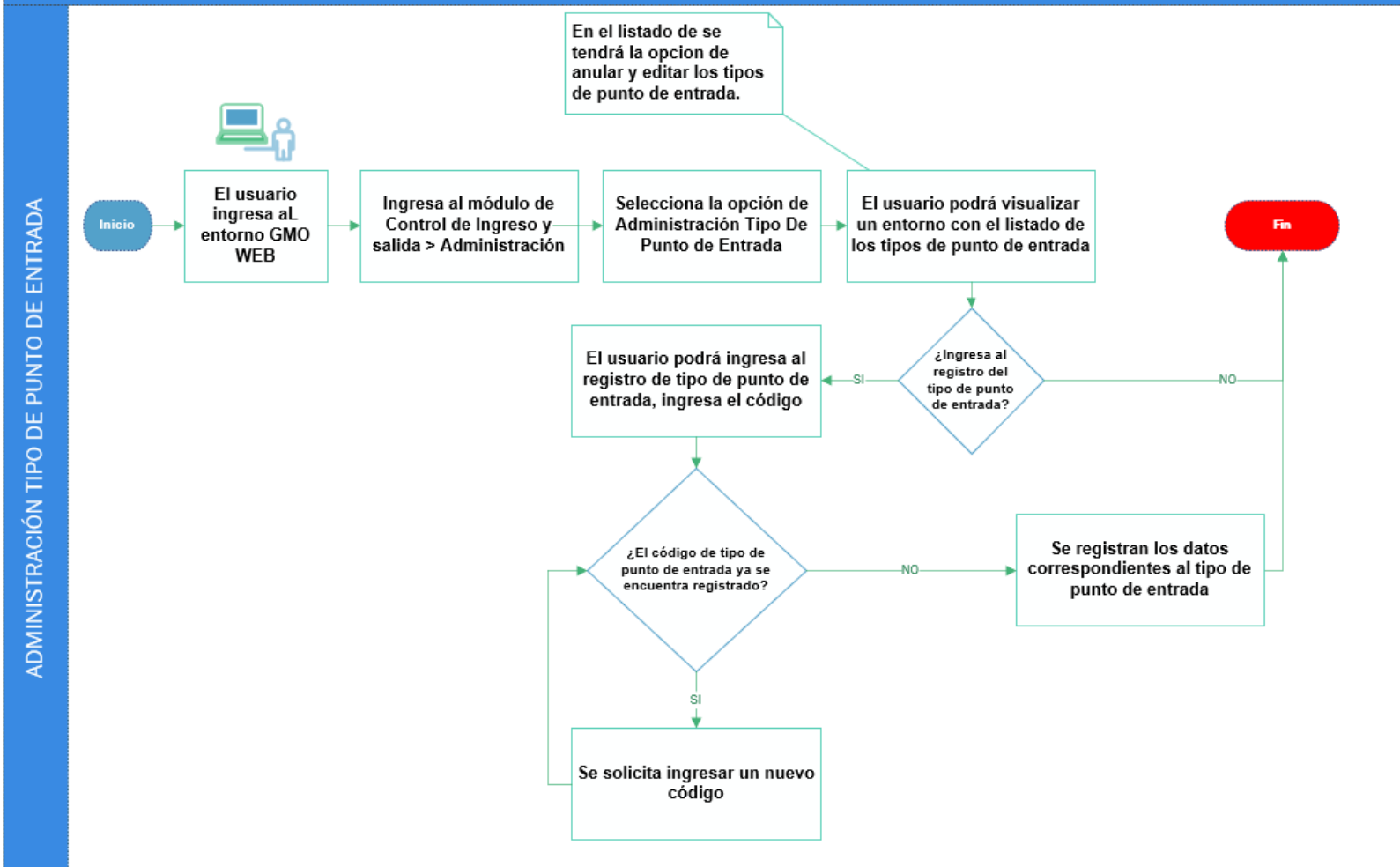


Figura 62. Diagrama BPMN de la Administración de Tipo de Punto de Entrada. Fuente: Elaboración propia.

CONTROL DE INGRESO Y SALIDA

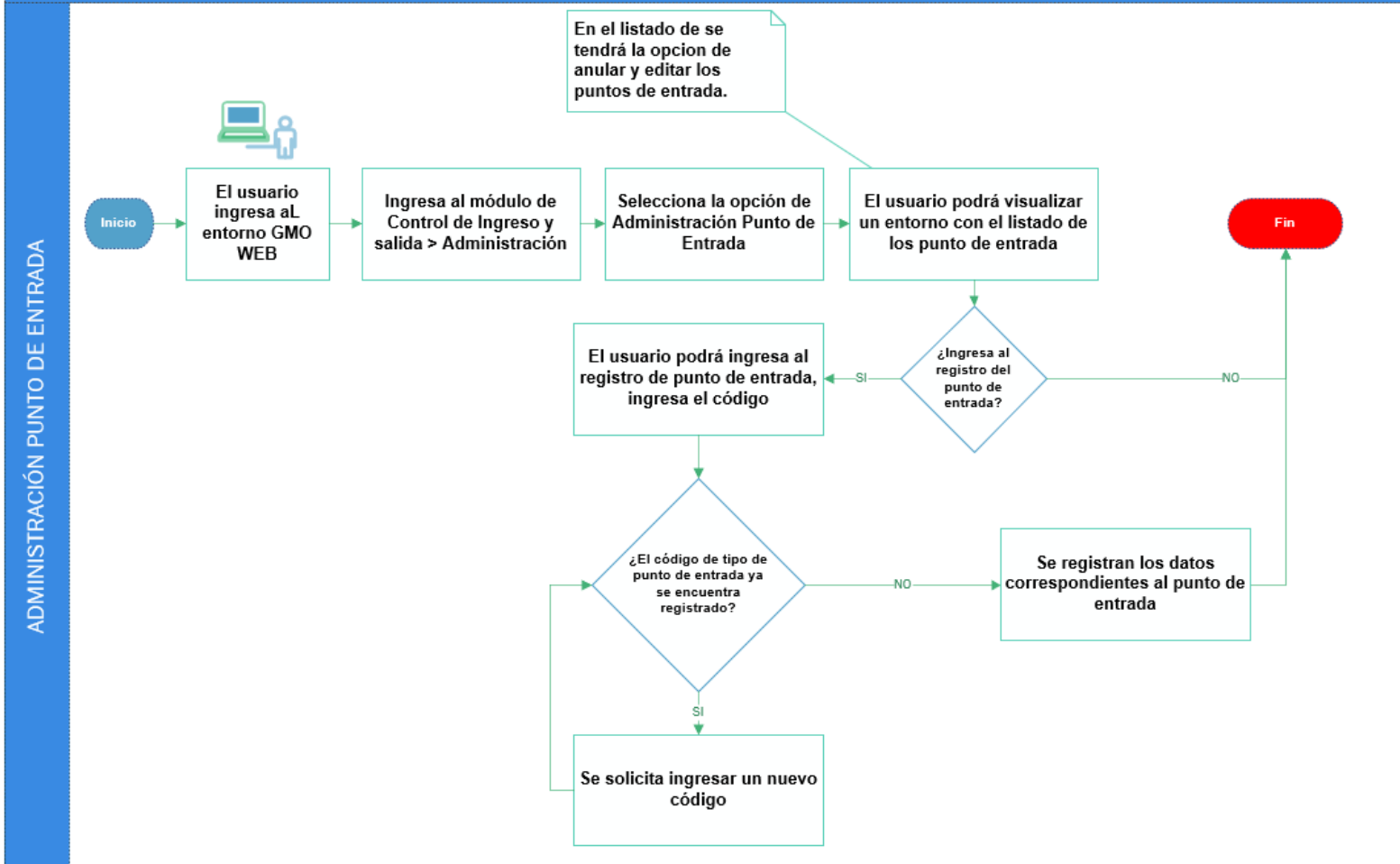


Figura 63. Diagrama BPMN de la Administración de Punto de Entrada. Fuente: Elaboración propia.

Para la conversión de la aplicación monolítica a microservicios se decidió migrar la administración del módulo de “Control de Ingreso y Salida” del aplicativo GMO, “Persona Observado”, “Punto de entrada” y “Tipo de punto de Entrada” servirán como caso práctico de la metodología MCM propuesta. Para ello el enfoque DDD permitirá desacoplar la aplicación teniendo conocimiento del contexto en base al dominio. Esto ayuda a comprender mas el negocio, el contexto acotado, teniendo un lenguaje ubicuo, conociendo las entidades, estructura y dominio del negocio. Asimismo, se evaluará el acoplamiento de los módulos, los cuales se independizarán en servicios independientes. Para ello realizamos una evaluación del acoplamiento y la cohesión.

Módulo de administración personal observado

Dentro del módulo de administración de personal observado se extrajeron 4 microservicios.

Microservicio de planilla

Durante el análisis del módulo de administración de personal observado se encontró que realizaba un consumo de una entidad llamada “planilla” esta entidad no contaba con una dependencia directa con el módulo, esto permitió poder evaluar la independencia de la entidad según la lógica del dominio negocio, se desarrolló una estructura independiente la cual será convertida en un microservicio, esto permitirá escalar las relaciones y/o características.

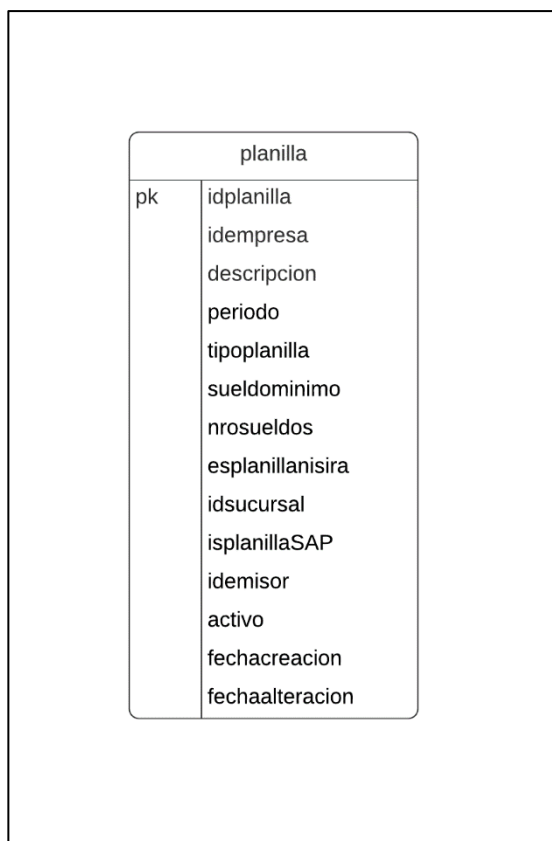


Figura 64. Diagrama relacional del microservicio de planillas. Fuente: Elaboración propia.

Microservicio de motivox

Durante el análisis del módulo de administración de personal observado se encontró que realizaba dos consumos dependientes entre las entidades “motivox” y “tipomotivox”, dichas entidades estaban relacionadas entre sí, pero no dependían directamente del módulo, esto permitió poder evaluar la independencia de las entidades en un solo microservicio, permitiendo escalar funcionalidades o características según la lógica del dominio de negocio. Asimismo, se desarrolló una estructura independiente la cual será convertida en un microservicio.

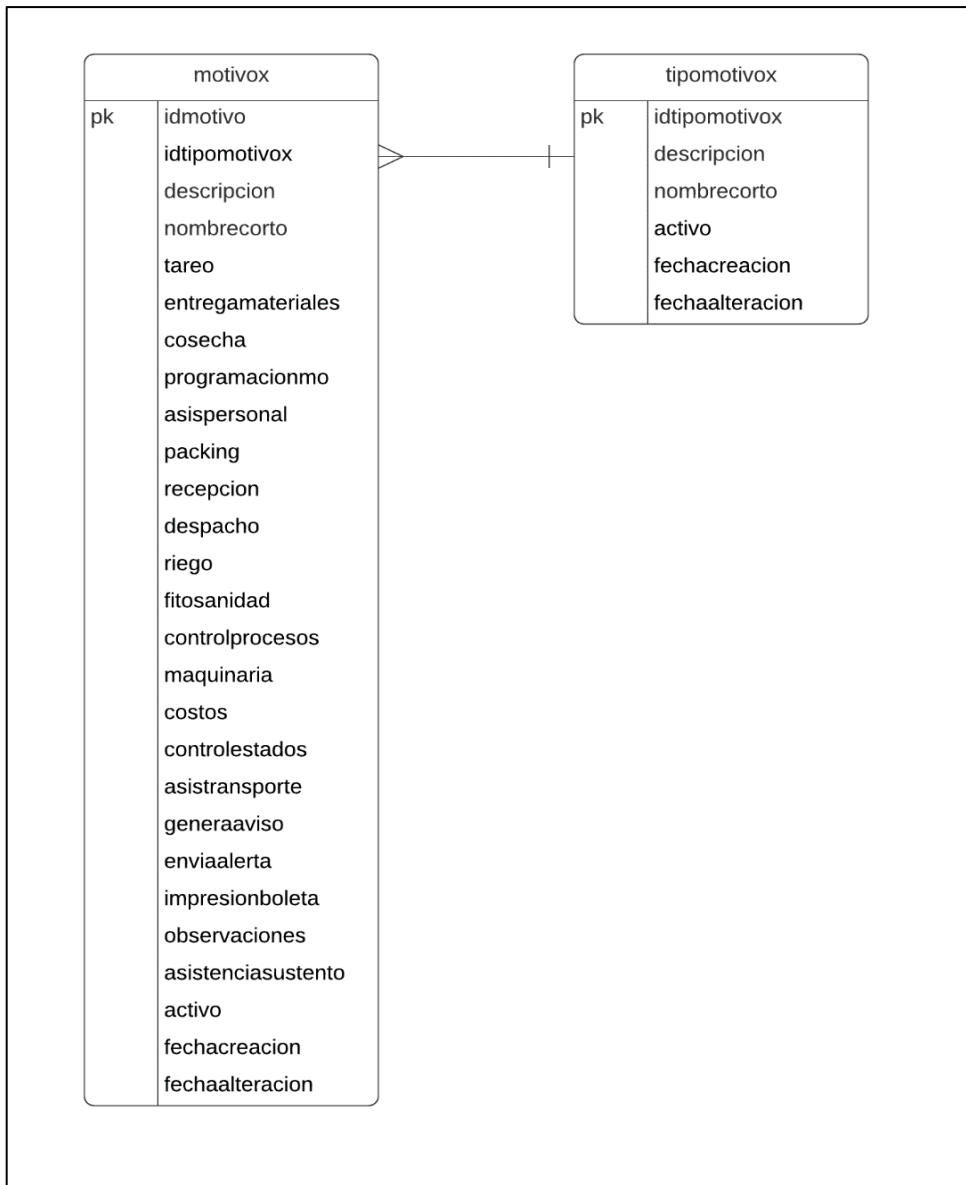


Figura 65. Diagrama relacional del microservicio de motivox. Fuente: Elaboración propia.

Microservicio de Trabajador

Durante el análisis del módulo de administración de personal observado se encontró que realizaba un consumo de la entidad “trabajador”, dicha entidad no dependía directamente del módulo, pero se usaba para comprobar los trabajadores existentes y realizar un volcado masivo de trabajadores a observar, esto permitió poder evaluar la independencia de las entidades en un solo microservicio, permitiendo escalar funcionalidades o características según la lógica del dominio de negocio. Asimismo, se desarrolló una

estructura independiente la cual será convertida en un microservicio.



Figura 66. Diagrama relacional del microservicio de trabajador. Fuente: Elaboración propia.

Microservicio de Trabajador observado

Durante el análisis del módulo de administración de personal observado se encontró que realizaba dos consumos, las entidades “trabajadorobservado” y “trabajadorobservadoxplanilla”, dichas entidad si eran dependientes del módulo, esto permitió evaluar las entidades e independizar las entidades del módulo para que sea un microservicio, permitiendo escalar funcionalidades o características según la lógica del dominio de negocio. Asimismo, se desarrolló una estructura independiente la cual será convertida en un microservicio.

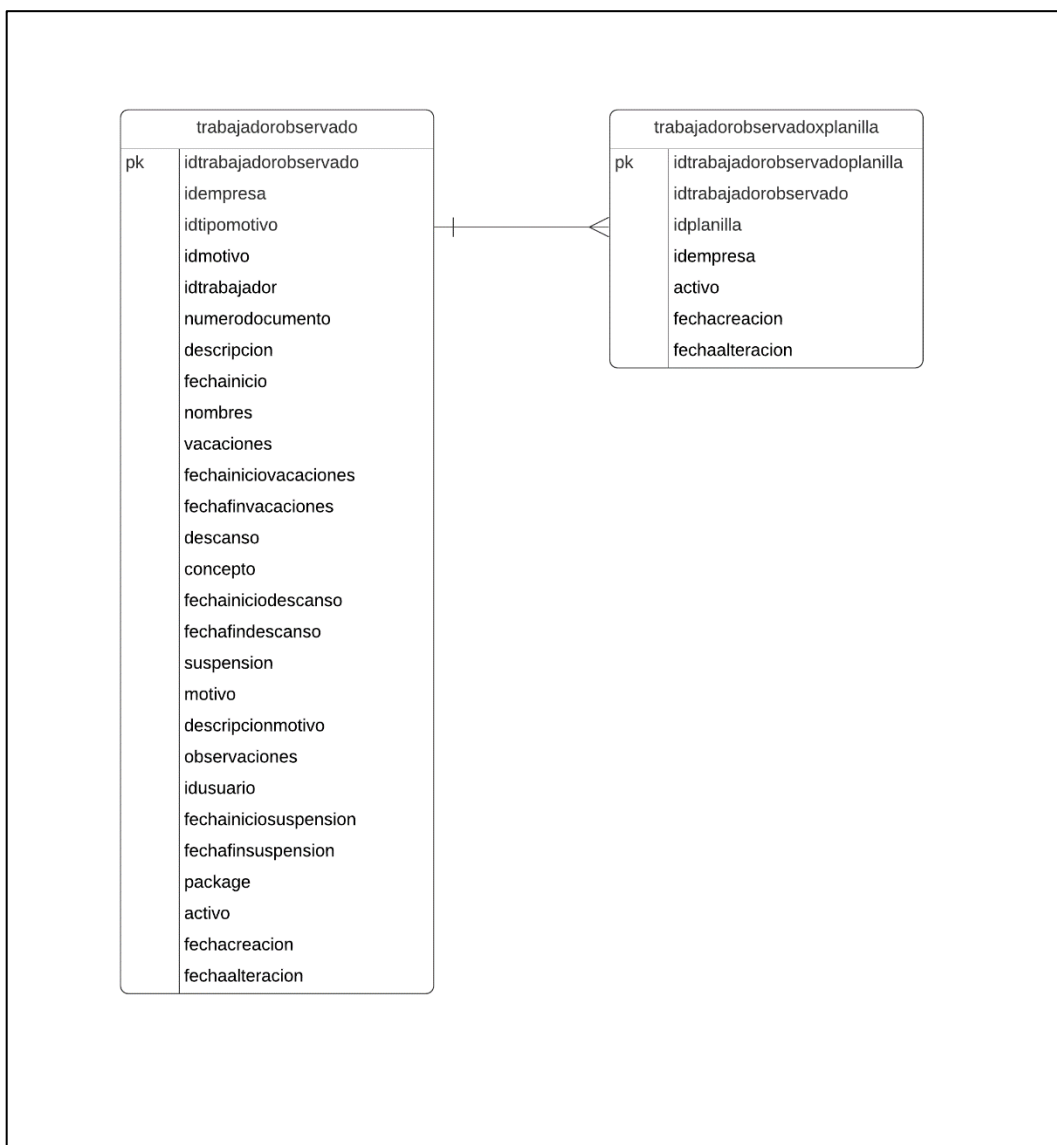


Figura 67. Diagrama relacional del microservicio de trabajador observado.
Fuente: Elaboración propia.

Módulo de administración punto de entrada y tipo de punto de entrada

Dentro del módulo de administración de punto de entrada y tipo de punto de entrada se extrajeron 1 microservicios.

Microservicio punto de entrada

Durante el análisis del módulo de administración de punto de entrada y tipo de punto de entrada se encontró que realizaba un consumo de dos entidades llamadas “puntoentrada” y “tipopuntoentrada” estas entidades aunque estaban en submódulos diferentes estaban relacionadas de manera dependiente, esto permitió poder evaluar la independencia de las entidades según la lógica del dominio negocio, logrando desarrollarse una estructura independiente la cual será convertida en un microservicio, esto permitirá escalar las relaciones y/o características.

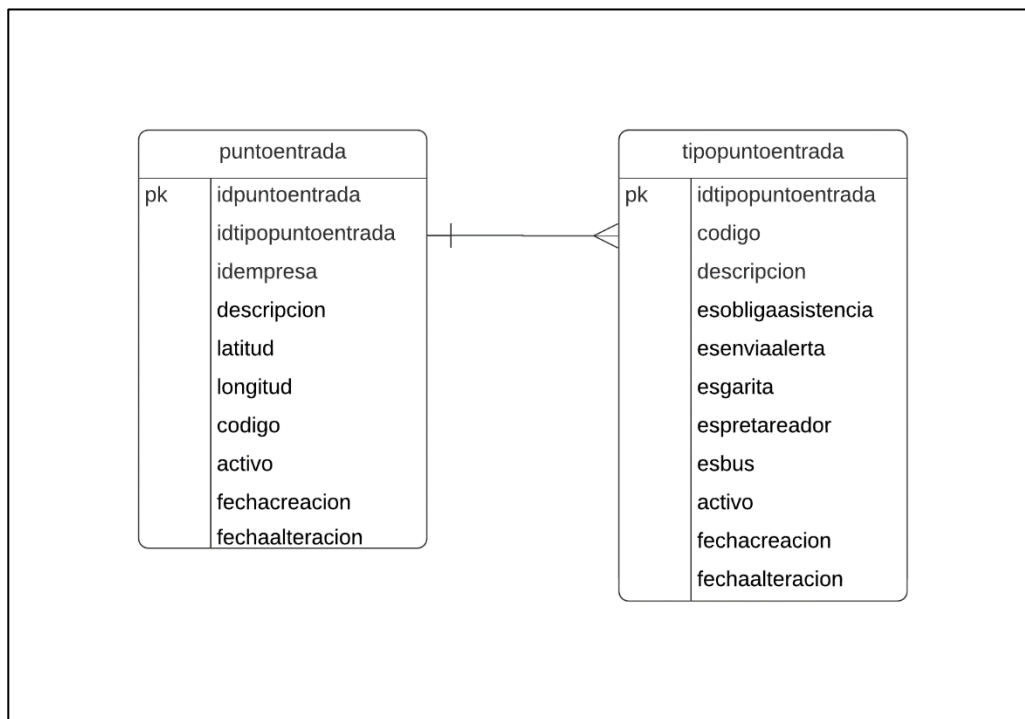


Figura 68. Diagrama relacional del microservicio de punto de entrada.
Fuente: Elaboración propia.

El análisis del dominio del negocio según el enfoque DDD permitió tener la perspectiva del desacoplamiento de la aplicación monolítica a 5 microservicios independientes, dichos microservicios se comunicarán con el cliente para lograr obtener la información necesaria según sea requerida.

D. Desarrollo de software

Durante el proceso de desarrollo de software se hizo uso de las historias de usuario, el análisis de las tecnologías, el enfoque DDD y se abstraio los microservicios a desarrollar. El esquema del desarrollo es poder comunicar el cliente (web, móvil) con los microservicios a través de una Api Gateway, encargándose de ser el único punto de entrada de todos los clientes, mejorando el acoplamiento, seguridad, comunicación entre microservicios, asimismo permitiría exponer API independientes para cada cliente adaptada a sus necesidades (Ma, Fan, et al., 2018).

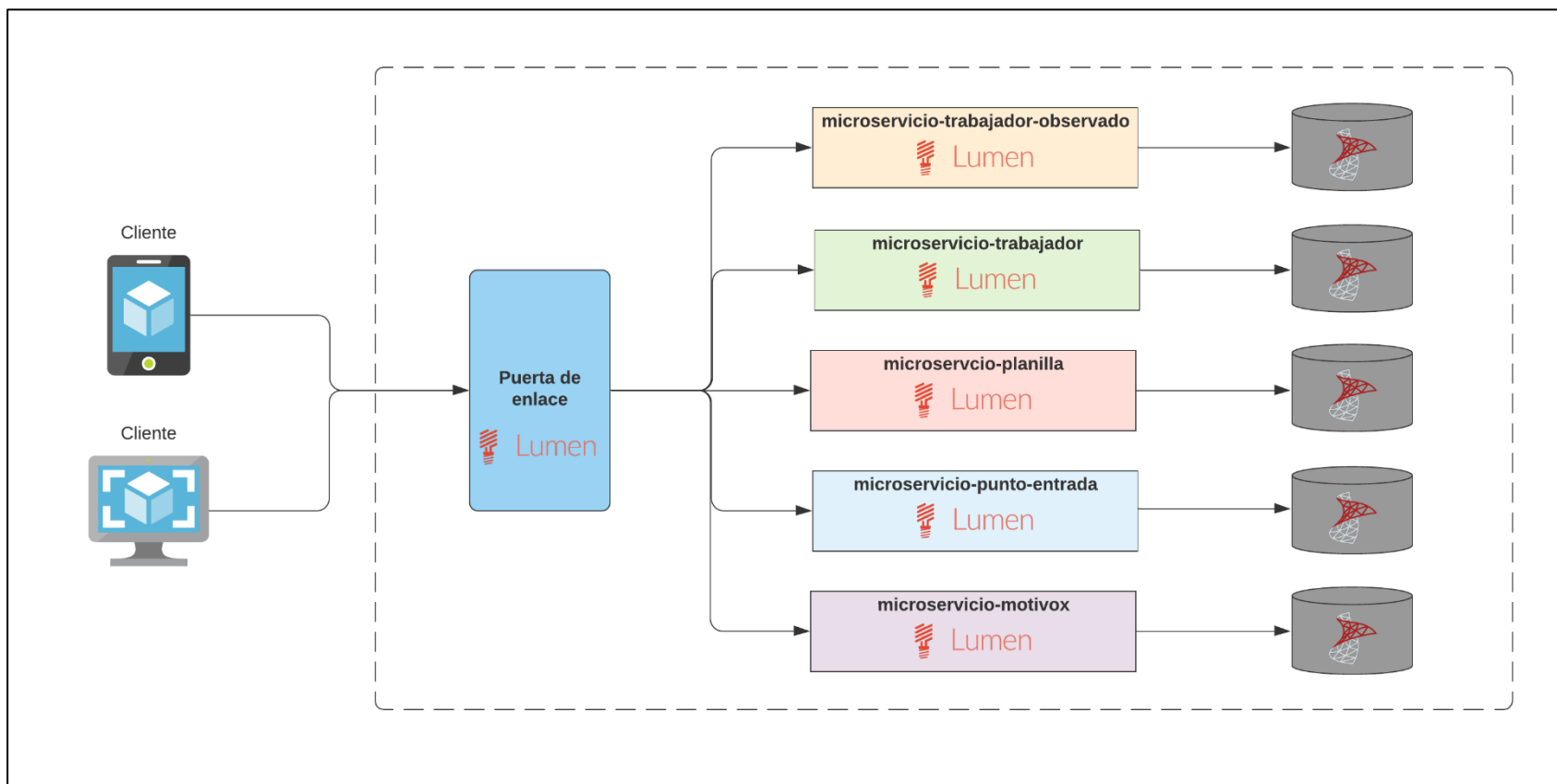


Figura 69. Diagrama representacional de la arquitectura microservicios del módulo de GMO. Fuente: Elaboración propia.

El desarrollo del cliente se hizo bajo el framework front-end VueJS, el Api Gateway y los microservicios se desarrollaron bajo del micro-framework de Lumen y para las bases de datos se utilizó SQLServer. La codificación se realizó en el IDE PhpStorm.

Para la comunicación entre el cliente y el Api Gateway se hizo uso del token JWT según el estándar RFC 7519 para permitir el intercambio de información de manera seguro. Además cuando se comunica el Api Gateway con los microservicios hace uso de una clave secreta de formato "<microservice>_SERVICE_SECRET", dicha clave sirve como segunda verificación de seguridad entre los microservicios (IETF, 2015).

Para llevar un mejor control de los microservicios a desarrollar se hizo uso de un VCS, esto permitió gestionar mejor los cambios realizados, brindando una línea de tiempo de las modificaciones realizadas y quien las realizó.

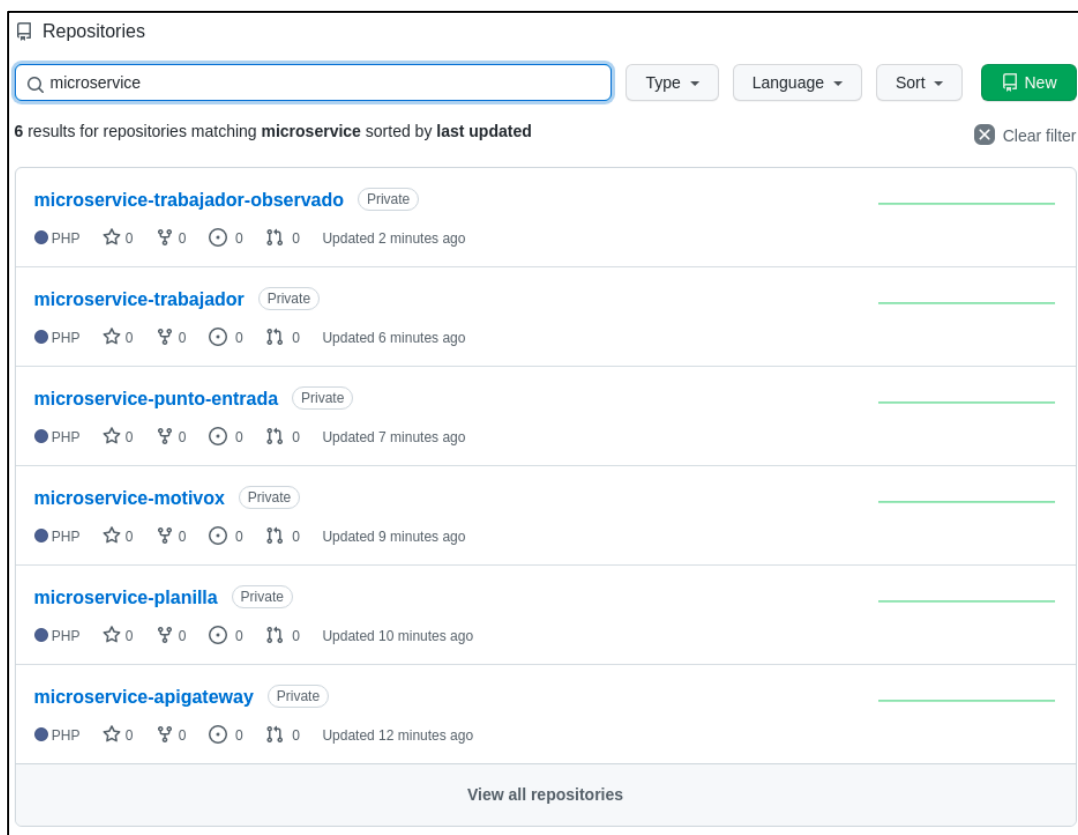


Figura 70. Repositorios de los microservicios alojados en Github de manera privada. Fuente: Elaboración propia.

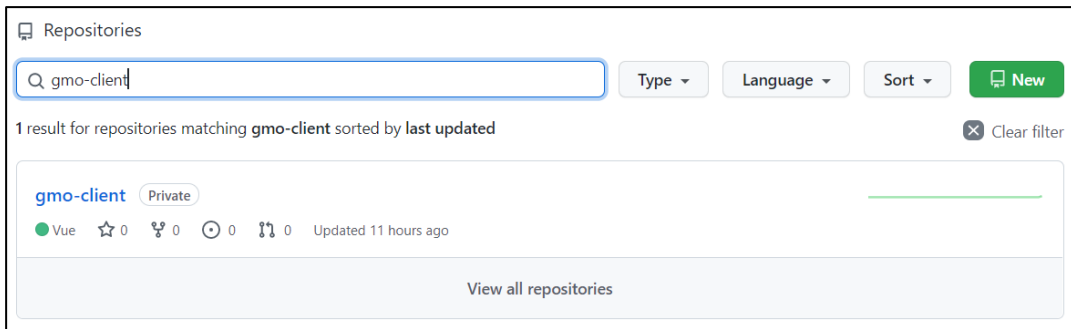


Figura 71. Repositorios del cliente web alojado en Github de manera privada.
Fuente: Elaboración propia.

Las rutas expuestas en el Api Gateway para que sea consumida por el cliente y que sirva de comunicaciones entre los microservicios se documentaron haciendo uso del siguiente formato.

Ficha de API'S					
Modulo/Microservicio	Funcionalidad	API	Metodo	Entrada	Salida
microservice_planilla	Listar planillas	/api/v1/planillas	GET	{}	{ data: [] }
microservice_motivox	Listar motivos	/motivox	GET	{ idTipoMotivo: "" }	{ data: [] }
	Listar tipos de motivos	/motivox/tipo	GET	{}	{ data: [] }
microservice_trabajador	Buscar trabajador por dni	/trabajador/search	GET	[{ dni: "" }]	{ data: [] }
microservice_punto_entrada	Listar puntos de entrada	/punto-entrada	GET	{}	
	Insertar puntos de entrada	/punto-entrada	POST	{ body: {} }	{ message: [] }
	Actualizar puntos de entrada	/punto-entrada	PUT	{ body: {} }	{ message: [] }
	Activar/Inactivar puntos de entrada	/punto-entrada/active	PATCH	{ Idpuntoentrada: "" }	{ message: [] }

	Verificar existencia de código de punto de entrada	/punto-entrada/check	GET	{ ldpuntoentrada: "" }	{ data: [] }
	Listar tipos de puntos de entrada	/tipo-punto-entrada	GET	{}	{ data: [] }
	Insertar tipo de punto de entrada	/tipo-punto-entrada	POST	{ body: {} }	{ message: [] }
	Actualizar tipo de punto de entrada	/tipo-punto-entrada	PUT	{ body: {} }	{ message: [] }
	Activar/Inactivar tipo de punto de entrada	/tipo-punto-entrada/active	PATCH	{ ldtipopuntoentrada: "" }	{ message: [] }
	Listar puntos de entrada	/tipo-punto-entrada/combo	GET	{}	{ data: [] }
	Verificar existencia de código de tipo ponto de entrada	/tipo-punto-entrada/check	GET	{ ldtipopuntoentrada: "" }	{ data: [] }
Microservice-trabajador-observado	Listar trabajadores observados	/trabajador-observado	GET	{}	{ data: [] }
	Insertar trabajadores observados	/trabajador-observado	POST	{ body: {} }	{ message: [] }

	Activar/Inactivar trabajadores observados	/trabajador-observado/active	PATCH	{ Idtrabajadorobservado: "" }	{ message: [] }
--	---	------------------------------	-------	-------------------------------------	-----------------------

Figura 72. Ficha de Apis para el consumo de los microservicios. Fuente: Elaboración propia.

Al haber realizado el desarrollo de los microservicios, del cliente web y el consumo correspondiente de las API's, se realizaron las pruebas correspondientes para las historias de usuario según los siguientes formatos:

Guía de pruebas						
Número y Nombre de Historia de usuario: H0001 > CIS – Administración de Punto de Entrada						
Personas involucradas: Jersson Mendoza Linares						
Ambiente de pruebas: Dev			Cronograma: S0001			
Observaciones: -----						
Casos de Prueba						
ID	Módulo	Descripción	Ruta	Resultado esperado	Resultado de desarrollo	Resultado de testing
1	Administración de Punto de entrada	Se debe de cargar el listado de puntos de entrada	control/administración/punto-entrada	Listado de puntos de entrada	Se listan correctamente los puntos de entrada	APROBADO
2	Administración de Punto de entrada	En el registro de punto de entrada se debe de validar el código	control/administración/punto-entrada	Mensaje de validación si el código ya existe	Si el código existe muestra el mensaje de "ya existe un punto de entrada con el código xxxx"	APROBADO
3	Administración de Punto de entrada	En el registro del punto de entrada se debe de cargar un listado con los tipos de punto de entrada	control/administración/punto-entrada	Listado de tipos de puntos de entrada	Se listan los tipos de punto de entrada	APROBADO
4	Administración de Punto de entrada	En el registrar se deben de validar que los datos registrados sean validos	control/administración/punto-entrada	El Punto de entrada se registre correctamente	Se registra correctamente devolviendo el mensaje de "Se registró correctamente"	APROBADO
5	Administración de Punto de entrada	En el editar de punto de entrada se debe de validar el código	control/administración/punto-entrada	Mensaje de validación si el código ya existe	Si el código existe muestra el mensaje de "ya existe un punto de entrada con el código xxxx"	APROBADO
6	Administración de Punto de entrada	En el editar del punto de entrada se debe de cargar un listado con los tipos de punto de entrada	control/administración/punto-entrada	Listado de tipos de puntos de entrada	Se listan los tipos de punto de entrada	APROBADO

7	Administración de Punto de entrada	En el editar se deben de validar que los datos registrados sean validos	control/administración/punto-entrada	El Punto de entrada se edita correctamente	Se edita correctamente devolviendo el mensaje de "Se actualizó correctamente"	APROBADO
8	Administración de Punto de entrada	Se permite anular un punto de entrada y volverlo a activar según sea el caso	control/administración/punto-entrada	El punto de entrada se anula o activa correctamente	Se anula o activa correctamente devolviendo el mensaje "se anuló correctamente el registro" o "se activó correctamente el registro"	APROBADO

Figura 73. Guía de pruebas de Punto de Entrada. Fuente: Elaboración propia.

Informe de Pruebas		
Número y Nombre de Historia de usuario: H0001 > CIS – Administración de Punto de Entrada		
Personas Involucradas: Jersson Mendoza Linares	Fecha: 01/10/2021	
Descripción: -----		
Pruebas	Número	Porcentaje
Casos de pruebas ejecutadas	8	100%
Casos de pruebas con éxito	8	100%
Casos de pruebas con error	0	0%
Casos de pruebas enviadas para la corrección	0	0%

Figura 74. Informe de pruebas de Punto de Entrada. Fuente: Elaboración propia.

Guía de pruebas						
Número y Nombre de Historia de usuario: H0002 > CIS – Administración de Tipos de Punto de Entrada						
Personas involucradas: Jersson Mendoza Linares						
Ambiente de pruebas: Dev			Cronograma: S0001			
Observaciones: -----						
Casos de Prueba						
ID	Módulo	Descripción	Ruta	Resultado esperado	Resultado de desarrollo	Resultado de testing
1	Administración de Tipo de Punto de entrada	Se debe de cargar el listado de tipos de puntos de entrada	/control/administracion/tipo-punto-entrada	Listado de tipos de puntos de entrada	Se listan correctamente los tipos de puntos de entrada	APROBADO
2	Administración de Tipo de Punto de entrada	En el registro de tipos de punto de entrada se debe de validar el código	/control/administracion/tipo-punto-entrada	Mensaje de validación si el código ya existe	Si el código existe muestra el mensaje de "ya existe un tipo de punto de entrada con el código xxxx"	APROBADO
3	Administración de Tipo de Punto de entrada	En el registrar se deben de validar que los datos registrados sean validos	/control/administracion/tipo-punto-entrada	El Tipo de Punto de entrada se registre correctamente	Se registra correctamente devolviendo el mensaje de "Se registró correctamente "	APROBADO
4	Administración de Tipo de Punto de entrada	En el editar de tipo de punto de entrada se debe de validar el código	/control/administracion/tipo-punto-entrada	Mensaje de validación si el código ya existe	Si el código existe muestra el mensaje de "ya existe un tipo de punto de entrada con el código xxxx"	APROBADO
5	Administración de Tipo de Punto de entrada	En el editar se deben de validar que los datos registrados sean validos	/control/administracion/tipo-punto-entrada	El Tipo de Punto de entrada se edita correctamente	Se edita correctamente devolviendo el mensaje de "Se actualizó correctamente "	APROBADO
6	Administración de Tipo de Punto de entrada	Se permite anular un tipo de punto de entrada y volverlo a activar según sea el caso	/control/administracion/tipo-punto-entrada	El tipo de punto de entrada se anula o activa correctamente	Se anula o activa correctamente devolviendo el mensaje "se anuló correctamente el registro" o "se activó correctamente el registro"	APROBADO

Figura 75. Guía de pruebas de Tipo de Punto de Entrada. Fuente: Elaboración propia.

Informe de Pruebas		
Número y Nombre de Historia de usuario: H0002 > CIS – Administración de Tipos de Punto de Entrada		
Personas Involucradas: Jersson Mendoza Linares	Fecha: 05/10/2021	
Descripción: -----		
Pruebas	Número	Porcentaje
Casos de pruebas ejecutadas	6	100%
Casos de pruebas con éxito	6	100%
Casos de pruebas con error	0	0%
Casos de pruebas enviadas para la corrección	0	0%

Figura 76. Informe de pruebas de Tipo de Punto de Entrada. Fuente: Elaboración propia.

Guía de pruebas						
Número y Nombre de Historia de usuario: H0003 > CIS – Administración de Personal Observado y BL						
Personas involucradas: Jersson Mendoza Linares						
Ambiente de pruebas: Dev			Cronograma: S0001			
Observaciones: -----						
Casos de Prueba						
ID	Módulo	Descripción	Ruta	Resultado esperado	Resultado de desarrollo	Resultado de testing
1	Administración de Personal Observado	Se debe de cargar el listado de personal observado	/control/administracion/personal-observado	Listado de personal observado	Se listan correctamente los trabajadores observados	APROBADO
2	Administración de Personal Observado	En el registro de personal observado se cargan las planillas	/control/administracion/personal-observado	Listado de planillas	Se listan las planillas	APROBADO
3	Administración de Personal Observado	En el registro de personal observado se cargan los motivos	/control/administracion/registro-personal-observado	Listado de motivos	Se listan los motivos	APROBADO
4	Administración de Personal Observado	En el registro de personal observado se cargan los tipos de motivos	/control/administracion/registro-personal-observado	Listado de tipos de motivos	Se listan los tipos de motivos	APROBADO
5	Administración de Personal Observado	Se realiza una búsqueda de trabajador a través de un importar y un input de texto	/control/administracion/registro-personal-observado	Se realiza la búsqueda al trabajador si existe mediante un input o un importar	Si se encuentra el trabajador devuelve su nombre, si no lo encuentra devuelve "trabajador no encontrado"	APROBADO
6	Administración de Personal Observado	En el registrar se deben de validar que los datos registrados sean validos	/control/administracion/registro-personal-observado	Los trabajadores observados se registran correctamente	Se registra correctamente devolviendo el mensaje de "Se registró correctamente"	APROBADO
7	Administración de Personal Observado	Se permite anular un personal observado	/control/administracion/personal-observado	El personal observado se anula correctamente	Se anula correctamente devolviendo el mensaje "se anuló correctamente el registro"	APROBADO

Figura 77. Guía de pruebas de Personal Observado. Fuente: Elaboración propia.

Informe de Pruebas		
Número y Nombre de Historia de usuario: H0003 > CIS – Administración de Personal Observado y BL		
Personas Involucradas: Jersson Mendoza Linares	Fecha: 09/10/2021	
Descripción: -----		
Pruebas	Número	Porcentaje
Casos de pruebas ejecutadas	7	100%
Casos de pruebas con éxito	7	100%
Casos de pruebas con error	0	0%
Casos de pruebas enviadas para la corrección	0	0%

Figura 78. Informe de pruebas de Personal Observado. Fuente: Elaboración propia.

Luego de haber consumidas las Apis expuestas y haber realizados las pruebas correspondientes para los microservicios por el cliente web GMO++ el módulo de “Control de Ingreso y Salida” la administración del “Persona Observado”, “Punto de entrada” y “Tipo de punto de Entrada” se visualiza de la siguiente manera.

GMO++ Administración de personal observado

Control de Ingreso y Salida > Administración > Personal Observado

Fechas: 01-09-2021 ~ 30-09-2021 | Planilla: EMITIDO POR TEST | Estado: TODOS

6/983 Personal observado

ADM. PERSONAL OBSERVADO

<input type="checkbox"/>	ID PLANILLA	TIPO MOTIVO	NUMERO DOCUMENTO TRABAJADOR	NOMBRE COMPLETO	FECHA INICIO
<input type="checkbox"/>	EM1	ALERTA	001036681	Trabajador No Registrado	2021-09-10
<input type="checkbox"/>	EM1	ALERTA	00122819	Trabajador No Registrado	2021-09-10
<input type="checkbox"/>	EM1	ALERTA	001658414	Trabajador No Registrado	2021-09-10
<input type="checkbox"/>	EM1	ALERTA	001668414	Trabajador No Registrado	2021-09-10
<input type="checkbox"/>	EM1	ALERTA	001036682	Trabajador No Registrado	2021-09-10
<input type="checkbox"/>	EM1	ALERTA	00111320	Trabajador No Registrado	2021-09-10

Rows per page: 10 | 1-6 of 6

Copyright © 2021 – Innovahtec Solutions

Figura 79. Interfaz de listado del personal observado consumiendo el microservicio de planilla. Fuente: Elaboración propia.

Innovahtec

Control de Ingreso y Salida > Administración > Personal Observado > Registro Personal Observado

Registrar Personal Observado

Fecha Inicio: 2021-11-14 Fecha Fin: 2021-11-14

Planilla: EMITIDO POR TEST Tipo Motivo: SUSPENSION Motivo: SUSPENSION DE LABORES - OPERACIONES

DNI

DNI	TRABAJADOR	ACCIÓN
11111111	Trabajador no Identificado	
22222222	MENDOZA MENDOZA, GERMAN	

Detalle de la Observación
DETALLE

Copyright © 2021 – Innovahtec Solutions

Figura 80. Interfaz de registro del personal observado consumiendo el microservicio de planilla y motivox. Fuente: Elaboración propia.

Control de Ingreso y Salida > Administración > Personal Observado

Fechas: 01-09-2021 ~ 30-09-2021 | Planilla: EMITIDO POR TEST | Estado: TODOS

6/983 Personal observado

ADM. PERSONAL OBSERVADO

	ID PLANILLA	TIPO MOTIVO	NUMERO DOCUMENTO TRABAJADOR	NOMBRE COMPLETO	FECHA INICIO
<input checked="" type="checkbox"/>	EM1	ALERTA	001036681	Trabajador No Registrado	2021-09-10
<input type="checkbox"/>	EM1	AL	00122819	Trabajador No Registrado	2021-09-10
<input type="checkbox"/>	EM1	ALERTA	001658414	Trabajador No Registrado	2021-09-10
<input type="checkbox"/>	EM1	ALERTA	001668414	Trabajador No Registrado	2021-09-10
<input type="checkbox"/>	EM1	ALERTA	001036682	Trabajador No Registrado	2021-09-10
<input type="checkbox"/>	EM1	ALERTA	00111320	Trabajador No Registrado	2021-09-10

Rows per page: 10 | 1-6 of 6

Copyright © 2021 – Innovahtec Solutions

Figura 81. Interfaz de listado de personal observado donde se puede inactivar a un personal observado. Fuente: Elaboración propia.

GMO++ Administración de punto de entrada

Control de Ingreso y Salida > Administración > Punto Entrada

Estado: TODOS

ADM. PUNTO DE ENTRADA | Buscar | Fijar columnas | 28

CODIGO	DESCRIPCION	DESCRIPCION TIPO PUNTO ENTRADA	ACTIVO
001	Garita 1	Entrada Normal	Activo
002	Garita 2	Entrada por BUS	Activo
003	Packing	Entrada Normal	Activo
004	Packing - Esparrago	Entrada Interna - Packing	Inactivo
005	Packing - Uva	Entrada Interna - Packing	Activo
006	Packing - Arandano	Entrada Interna - Packing	Activo
007	Packing - Otros	Entrada Interna - Packing	Activo
008	Pruea hoy cds xD	Entrada Interna - Packing	Activo
009	RONDA 01	Entrada Tipo Garita	Activo
010	RONDA 02	Entrada Tipo Garita	Activo

Rows per page: 10 | 1-10 of 28

Copyright © 2021 – Innovahtec Solutions

Figura 82. Interfaz de listado de punto de entrada. Fuente: Elaboración propia.

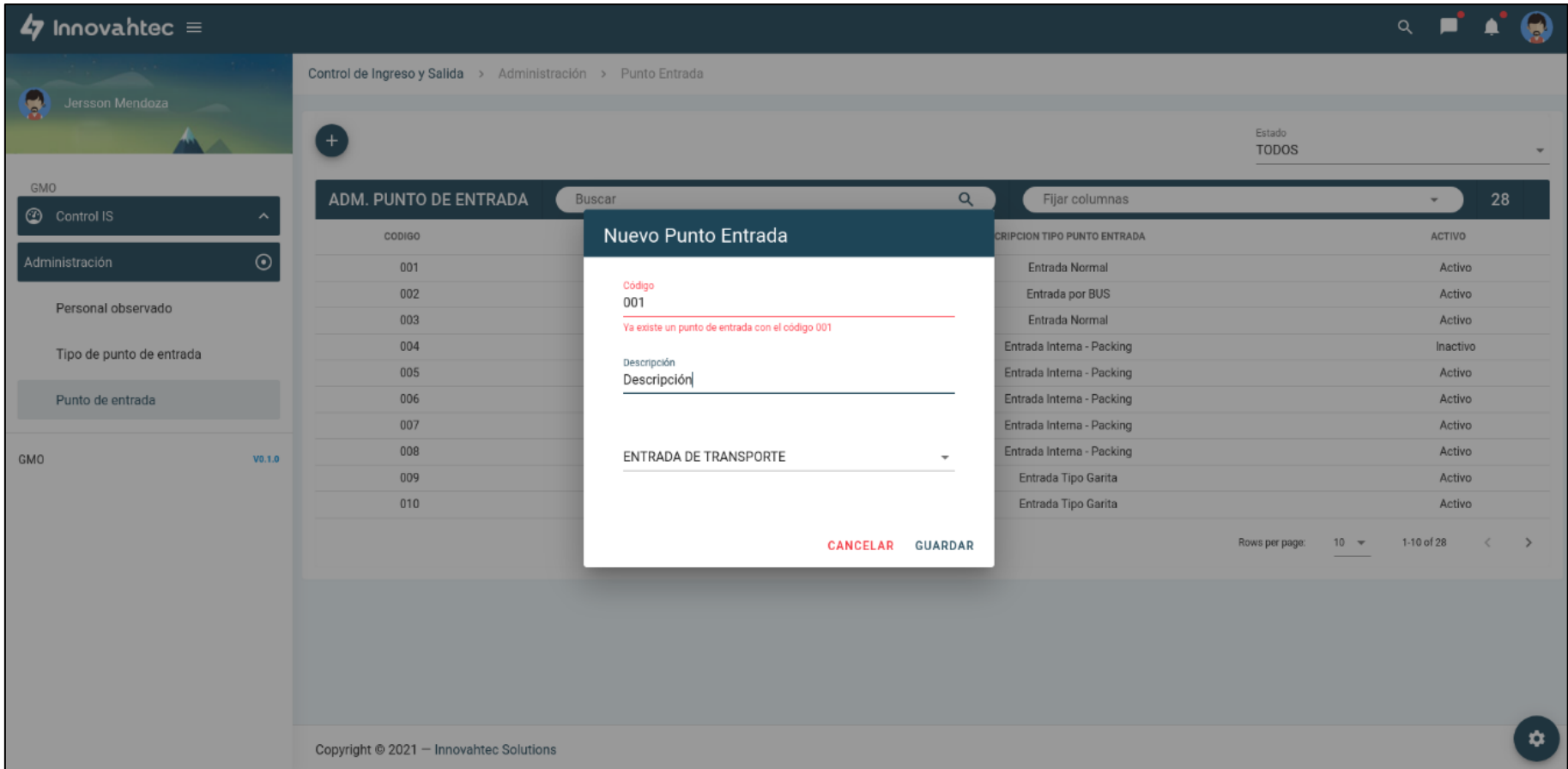


Figura 83. Interfaz de registro de punto de entrada donde se consumen los tipos de punto de entrada y su validación correspondiente. Fuente: Elaboración propia.

Control de Ingreso y Salida > Administración > Punto Entrada

Estado: TODOS

ADM. PUNTO DE ENTRADA

CODIGO	DESCRIPCION	DESCRIPCION TIPO PUNTO ENTRADA	ACTIVO
001	Garita 1	Entrada Normal	Activo
002	Garita 2	Entrada por BUS	Activo
003	Packing	Entrada Normal	Activo
004	Packing - Esparr...	Entrada Interna - Packing	Inactivo
005	Packing - Uva	Entrada Interna - Packing	Activo
006	Packing - Arandano	Entrada Interna - Packing	Activo
007	Packing - Otros	Entrada Interna - Packing	Activo
008	Pruea hoy cds xD	Entrada Interna - Packing	Activo
009	RONDA 01	Entrada Tipo Garita	Activo
010	RONDA 02	Entrada Tipo Garita	Activo

Rows per page: 10 1-10 of 28

Copyright © 2021 – Innovahtec Solutions

Figura 84. Interfaz de listado de punto de entrada donde se puede editar y anular un punto de entrada. Fuente: Elaboración propia.

GMO++ Administración de tipo punto de entrada

Control de Ingreso y Salida > Administración > Tipo Punto Entrada

Estado: TODOS

ADM. TIPO PUNTO ENTRADA

Buscar

Fijar columnas

1/21

CODIGO	DESCRIPCION	ES ENVIA ALERTA	ES GARITA	ES BUS	ES PRETAREADOR	ES OBLIGATORIO
ENT	ENTRADA DE TRANSPORTE	true	true	true	true	1
BUS	Entrada por BUS	true	false	true	true	
GAR	Entrada Tipo Garita	true	true	false	false	1
NOR	Entrada Normal	true	false	false	true	1
PAC	Entrada Interna - Packing	true	false	false	true	
PTC	Puntos de Control FIPE	false	false	false	false	1
ERT	ENTRADA DE TRANSPORTE upd	false	false	false	true	1
NTT	prueba	true	true	true	true	
ENC	ENTRADA DE CARRUSEL	false	true	false	false	1
TTT	prueba ultima	true	false	false	true	

Rows per page: 10 1-10 of 21

Copyright © 2021 – Innovahtec Solutions

Figura 85. Interfaz de listado de tipo de punto de entrada. Fuente: Elaboración propia.

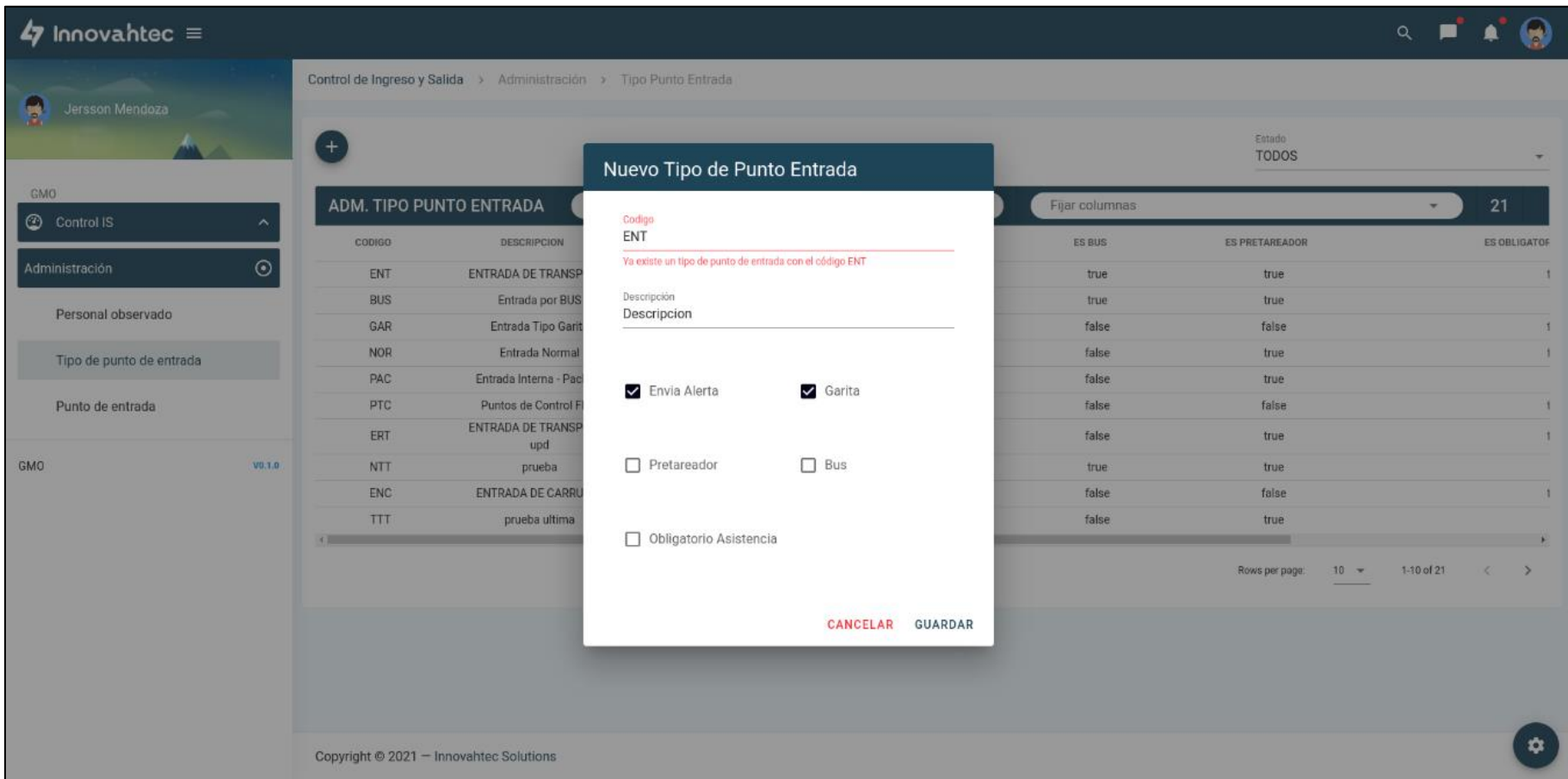


Figura 86. Interfaz de registro de punto de entrada con su validación correspondiente. Fuente: Elaboración propia.

The screenshot displays the 'ADM. TIPO PUNTO ENTRADA' interface. The table lists various entry point types with their respective attributes. A context menu is visible over the first row (ENT), providing options to edit or delete the entry point type.

CODIGO	DESCRIPCION	ES ENVIA ALERTA	ES GARITA	ES BUS	ES PRETAREADOR	ES OBLIGATORIO
ENT	ENTRADA DE TRANSPORTE	true	true	true	true	1
BUS	Entrada por BUS	true		true	true	
GAR	Entrada Tipo Garita	true		false	false	1
NOR	Entrada Normal	true		false	true	1
PAC	Entrada Interna - Packing	true	false	false	true	
PTC	Puntos de Control FIPE	false	false	false	false	1
ERT	ENTRADA DE TRANSPORTE upd	false	false	false	true	1
NTT	prueba	true	true	true	true	
ENC	ENTRADA DE CARRUSEL	false	true	false	false	1
TTT	prueba ultima	true	false	false	true	

Figura 87. Interfaz de listado de tipo de punto de entrada donde se puede editar y anular un tipo de punto de entrada. Fuente: Elaboración propia.

E. Despliegue de software

El despliegue del software se realizó a través de contenedores Docker, creando imágenes de los microservicios, para ello se creó un dockerfile para poder crear una imagen personalizada e instalar las dependencias y configuraciones necesarias para los microservicios y para la conexión a sus bases de datos respectivas mediante SQL Server.

```
~/microservices/project
) docker images | grep microservice
microservice-trabajador-observado 1.0 fae0a9c20775 33 seconds ago 552MB
microservice-trabajador 1.0 9577afe02a77 About a minute ago 552MB
microservice-punto-entrada 1.0 2522bb850699 2 minutes ago 553MB
microservice-motivox 1.0 0281194d539d 3 minutes ago 552MB
microservice-planilla 1.0 c49bf7a084b6 7 minutes ago 552MB
microservice-apigateway 1.0 92e53b7bd58c 11 minutes ago 544MB
```

Figura 88. Imágenes de microservicios con su versión correspondiente para contenedorizarlas a través de Docker. Fuente: Elaboración propia.

Para finalizar con la MCM el despliegue a través de un servicio cloud es de vital importancia para la disponibilidad del aplicativo. En este caso se eligió Digital Ocean teniendo en cuenta su agilidad, elasticidad, coste de infraestructura, calidad precio y que cumpla los requisitos de la ficha cloud. Digital ocean permite tener recursos escalables, elásticos, además de sus costes accesibles y velocidad de transferencia a la altura de los principales servicios cloud.

Ficha de Servicio Cloud		
Nombre del servicio: Servicio Cloud para microservicios		
Tipo de Servicio: Cloud	Costo de servicio: \$40/mes	
Descripción: Digital ocean contendra los microservicios desarrollados		
¿Cuenta con servicios ?	SI	NO
Load Balancer	X	
Container	X	
Orchestrator	X	
Continuous Deployment	X	

Figura 89. Ficha de servicio cloud. Fuente: Elaboración propia.

Evaluación de la propuesta metodológica mediante juicio de expertos.

Según el metodo general propuesto para la MCM se realizó una evaluación de juicio de expertos, los cuales tengan relación en el área de TI, permitiendo valorar la MCM propuesta en esta investigación. Asimismo, se decidió hacer uso del método Delphi.

Para la correcta aplicación del método Delphi se definió los expertos a ser consultados, para la mejor comprensión de la MCM se realizó un resumen de la propuesta haciendo énfasis en el proceso de la elaboración de la metodología y su modelo general hasta el diagrama metodológico de la MCM esto se puede observar en el **Anexo 3.1**.

Realizado el resumen de la MCM se identificó y selecciono a los expertos que participarían en la evaluación de la MCM, ellos cuentan con una experiencia en TI, esto se puede observar en el **Anexo 3.2**, los expertos fueron los siguientes:

- a) Ing. Denny Fuentes Adrianzén.
- b) Ing. Junior Eugenio Cachay Maco.
- c) Ing. Carlos Valdivia Salazar.

Seleccionados los expertos se decidió contactarlos para confirmar su apoyo en la evaluación de la MCM, una vez confirmada su participación se procedió a enviar un correo adjuntando el resumen de la MCM y un cuestionario de 10 preguntas que ayudarán con la evaluación de la metodología, como se puede observar en el **Anexo 3.3**. El método Delphi se pudo aplicar a través del correo electrónico gracias a su flexibilidad, esto permitió validar mediante juicio de expertos a la hipótesis de la investigación, dichas evidencias de la comunicación con los expertos se muestran en el **Anexo 3.4**.

La primera iteración realizada a los expertos se muestra de la siguiente manera.

Tabla 18.

Respuestas de la primera iteración del método Delphi.

Preguntas	Evaluador 1		Evaluador 2		Evaluador 3	
	Rpta	Comentario	Rpta	Comentario	Rpta	Comentario
1.- Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya analizado las características de las pequeñas empresas para poder plantear la metodología MCM de acorde a dichas características?	SI	Es correcto, porque el conocer las características va a permitir saber de primera fuente las necesidades reales que se deben de tener en cuenta y a la vez definir la delimitación del alcance de la propuesta.	SI	Porque se necesita establecer un contexto general para una propuesta ad hoc, basado en un alcance claro, tal como es una pequeña empresa.	SI	Es necesario identificar las tecnologías con las que cuentan los aplicativos monolíticos.
2.- Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya realizado un análisis de la literatura científica	SI	Es correcto, un punto clave y de partida es poder recopilar y analizar los estudios científicos, para que de esta manera se conozca la estructura de una arquitectura orientada a microservicios y en	SI	Porque la literatura científica es validada por académicos de alto nivel y presentan cuidado en la calidad de producción científica.	SI	Es necesario conocer el ámbito actual de las metodologías que tratan sobre el tema de microservicios, y poder realizar una nueva propuesta.

para extraer las características, herramientas más utilizadas en la arquitectura basada en microservicios y así poder tener un concepto general para el desarrollo de la MCM propuesta?

función a ello plantear una alternativa acorde a las necesidades identificadas.

3.- Según la metodología MCM propuesta.
¿Considera usted correcto que la elaboración del modelo metodológico describe mediante las actividades a realizar, un nivel de abstracción y relación comprensible para todo el proceso de la metodología MCM propuesta?

SI

Es correcto, en el modelo planteado se define todo el conjunto de actividades y tareas que deben de ejecutarse que permitan identificar las características propias que se sigue en el proceso de la metodología en todo su ciclo de vida de desarrollo.

SI

Porque estás considerando verificación por expertos y validación con un caso de estudio.

SI

Abarca todas las etapas necesarias para descomponer y desacoplar la aplicación monolítica en una serie de microservicios.

4.- Según la metodología MCM propuesta.
¿Considera usted correcto que la elaboración de un análisis de factibilidad de migración de una arquitectura monolítica a una basada en microservicios basado en un cuestionario puede establecer un juicio correcto para validar la “factibilidad de migración” en la metodología MCM propuesta?

SI

Es Correcto, el análisis de la factibilidad va a permitir conocer si es posible hacer el proceso de migración conociendo los aspectos técnicos, económicos y operativos a considerarse que garanticen la fiabilidad del modelo a desarrollar.

NO

Porque estás considerando que el análisis lo realice un Gerente con sólo preguntas. Esto amerita que:
1. El Gerente tenga un alto nivel de experiencia en operatividad.
2. Se obvien algunos aspectos técnicos.

SI

Es necesario conocer si es factible convertir una aplicación monolítica a microservicios.

5.- Según la metodología MCM propuesta.
¿Considera usted correcto que la actividad de

SI

Es correcto, el objetivo de un paradigma ágil es poder reducir los tiempos de desarrollo y tener en contacto directo a los usuarios en cada

SI

Porque para pequeñas empresas se necesita agilidad.

SI

Como la metodología está basada en las fases de la ingeniería del software, es necesario el uso de una metodología ágil

planeamiento de software se haga uso de una metodología ágil como Scrum, XP, con el fin de hacer mejoras continuas en el desarrollo de software?

iteración que se tiene que realizar, teniendo en cuenta que un punto de partida es poder considerar las necesidades o requerimientos de usuarios.

para que acelere el proceso.

6.- Según la metodología MCM propuesta. ¿Considera usted que para la actividad de diseño de software es correcto hacer un análisis de la tecnología actual y la tecnología a migrar a través de un cuadro comparativo para tener un concepto general de las ventajas y desventajas de las mismas?

SI

Es correcto, la actividad de Diseño es uno de los puntos críticos, es decir, en ella se plantea el modelo de propuesta considerando la evaluación del paradigma actual y el propuesto, para ello debe de organizarse un comparativos de factores de evaluación.

SI

Porque orientará al proceso de reingeniería y en el uso de recursos sólidos ya existentes.

SI

Es necesario realizar este cuadro comparativo, sin embargo, es necesario que se cuente con expertos en el área para identificar estas ventajas y desventajas.

7.- Según la metodología MCM propuesta.
¿Considera usted correcto hacer uso de un modelado con base en BPMN, con la finalidad de tener un mejor concepto del proceso a realizar?

SI

Es correcto, considerar que el modelar los procesos de negocios nos va poder conocer la integridad y relacionamiento de cada uno de los elementos interactuantes del sistema, a la vez que se puede generar una simulación previa del modelo y poder evaluar sus efectos y sensibilidad como parte del sistema.

SI

Porque representa de manera práctica el flujo de información que sostendrá la solución tecnológica propuesta.

SI

El manejo de base de datos es importante en todo desarrollo de software, y en el caso de microservicios mucho más.

8.- Según la metodología MCM propuesta.
¿Considera usted correcto que hacer uso del enfoque Domain Driven Design (DDD) ayuda a los involucrados en el proceso de migración a manejar un mejor lenguaje ubicuo?

SI

Es correcto, El Diseño del Dominio, va a permitir definir claramente cuál es el alcance que debe de tenerse, además que con ello se puede delimitar correctamente los costos, tiempos y riesgos del desarrollo en base al modelo propuesto.

SI

Porque es esencial manejar un lenguaje uniforme de comunicación para reforzar el conocimiento de desarrollo, y evitar así problemas ante rotación o renuncia de personal del proyecto.

SI

Teniendo que DDD considera al conocimiento del dominio para crear una aplicación, es importante contar con un lenguaje entre las personas que conocen el proceso y los expertos.

<p>9.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI retiraría/adicionaría algún elemento del modelo metodológico, para generar una metodología aplicada a una pequeña empresa?</p>	<p>NO</p>	<p>Considerar que se está proponiendo un modelo y la idea es poder evaluar sus resultados y en función a ello, poder hacer los ajustes necesarios que garanticen un mejor soporte a los objetivos propuestos como alternativa de solución.</p>	<p>SI</p> <p>Porque aún falta mejorar actividades de verificación y validación de software (V&V), si bien no se necesita una gama amplia de pruebas V&V, al menos lo básico para una pequeña empresa.</p>	<p>NO</p> <p>Están consideradas las etapas necesarias para un modelo de migración de una aplicación monolítica a microservicios.</p>
<p>10.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI cree que la metodología MCM a través de sus 5 actividades disminuye la complejidad de migración de arquitectura a microservicios?</p>	<p>SI</p>	<p>Es correcto, es importante es poder conocer paso a paso cada una de las actividades y más aún si el proceso de abstracción es por módulos o etapas de esta manera se puede identificar cada uno de los detalles de los elementos intervinientes.</p>	<p>SI</p> <p>Porque abarca muchos aspectos importantes para la migración ordenada.</p>	<p>SI</p> <p>Está basada en un análisis de las tecnologías de la aplicación monolítica y la propuesta del uso de nuevas tecnologías hacia los microservicios.</p>

Fuente: Elaboración propia

Luego de realizar la primera iteración mediante el método Delphi a los expertos se ha mantenido en anonimato la identidad de los evaluadores. Habiendo remitido el resumen y el cuestionario correspondiente, las respuestas de la primera iteración servirán para poder reflexionar o reafirmar los puntos de vista en la segunda iteración, de la cual se evidencia la comunicación con los expertos en el **Anexo 3.4**.

Tabla 19.

Respuesta del evaluador 1 en la segunda iteración del método Delphi.

Preguntas	Evaluador 1		Evaluador 2		Evaluador 3		Segunda opinión
	Rpta	Comentario	Rpta	Comentario	Rpta	Comentario	
1.- Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya analizado las características de las pequeñas empresas para poder plantear la metodología MCM de acorde a dichas características?	SI	Es correcto, porque el conocer las características va a permitir saber de primera fuente las necesidades reales que se deben de tener en cuenta y a la vez definir la delimitación del alcance de la propuesta.	SI	Porque se necesita establecer un contexto general para una propuesta ad hoc, basado en un alcance claro, tal como es una pequeña empresa.	SI	Es necesario identificar las tecnologías con las que cuentan los aplicativos monolíticos.	Conservo mi opinión
2.- Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya realizado un	SI	Es correcto, un punto clave y de partida es poder recopilar y analizar los estudios científicos, para que de esta manera se conozca la	SI	Porque la literatura científica es validada por académicos de alto nivel y presentan cuidado en la calidad de producción	SI	Es necesario conocer el ámbito actual de las metodologías que tratan sobre el tema de microservicios, y poder realizar una nueva propuesta.	Conservo mi opinión

<p>análisis de la literatura científica para extraer las características, herramientas más utilizadas en la arquitectura basada en microservicios y así poder tener un concepto general para el desarrollo de la MCM propuesta?</p>	<p>estructura de una arquitectura orientada a microservicios y en función a ello plantear una alternativa acorde a las necesidades identificadas.</p>	<p>científica.</p>		
<p>3.- Según la metodología MCM propuesta. ¿Considera usted correcto que la elaboración del modelo metodológico describe mediante las actividades a realizar, un nivel de abstracción y relación comprensible para todo el proceso de</p>	<p>Es correcto, en el modelo planteado se define todo el conjunto de actividades y tareas que deben de ejecutarse que permitan identificar las características propias que se sigue en el proceso de la metodología en todo su ciclo de vida de desarrollo.</p>	<p>Porque estás considerando verificación por expertos y validación con un caso de estudio.</p>	<p>Abarca todas las etapas necesarias para descomponer y desacoplar la aplicación monolítica en una serie de microservicios.</p>	<p>Conservo mi opinión</p>

la metodología
MCM propuesta?

4.- Según la metodología MCM propuesta.
¿Considera usted correcto que la elaboración de un análisis de factibilidad de migración de una arquitectura monolítica a una basada en microservicios basado en un cuestionario puede establecer un juicio correcto para validar la “factibilidad de migración” en la metodología MCM propuesta?

SI

Es Correcto, el análisis de la factibilidad va a permitir conocer si es posible hacer el proceso de migración conociendo los aspectos técnicos, económicos y operativos a considerarse que garanticen la fiabilidad del modelo a desarrollar.

NO

Porque estás considerando que el análisis lo realice un Gerente con sólo preguntas. Esto amerita que:
1. El Gerente tenga un alto nivel de experiencia en operatividad.
2. Se obvien algunos aspectos técnicos.

SI

Es necesario conocer si es factible convertir una aplicación monolítica a microservicios.

Conservo mi opinión

5.- Según la metodología MCM propuesta.

SI

Es correcto, el objetivo de un paradigma ágil es

SI

Porque para pequeñas empresas se necesita agilidad.

SI

Como la metodología está basada en las fases

Conservo mi opinión

¿Considera usted correcto que la actividad de planeamiento de software se haga uso de una metodología ágil como Scrum, XP, con el fin de hacer mejoras continuas en el desarrollo de software?

6.- Según la metodología MCM propuesta.
¿Considera usted que para la actividad de diseño de software es correcto hacer un análisis de la tecnología actual y la tecnología a migrar a través de un cuadro comparativo para tener un concepto general de las ventajas y

SI

poder reducir los tiempos de desarrollo y tener en contacto directo a los usuarios en cada iteración que se tiene que realizar, teniendo en cuenta que un punto de partida es poder considerar las necesidades o requerimientos de usuarios.

Es correcto, la actividad de Diseño es uno de los puntos críticos, es decir, en ella se plantea el modelo de propuesta considerando la evaluación del paradigma actual y el propuesto, para ello debe de organizarse un comparativos de factores de evaluación.

SI

Porque orientará al proceso de reingeniería y en el uso de recursos sólidos ya existentes.

SI

de la ingeniería del software, es necesario el uso de una metodología ágil para que acelere el proceso.

Es necesario realizar este cuadro comparativo, sin embargo, es necesario que se cuente con expertos en el área para identificar estas ventajas y desventajas.

Conservo mi opinión

desventajas de las mismas?

7.- Según la metodología MCM propuesta.
¿Considera usted correcto hacer uso de un modelado con base en BPMN, con la finalidad de tener un mejor concepto del proceso a realizar?

SI

Es correcto, considerar que el modelar los procesos de negocios nos va poder conocer la integridad y relacionamiento de cada uno de los elementos interactuantes del sistema, a la vez que se puede generar una simulación previa del modelo y poder evaluar sus efectos y sensibilidad como parte del sistema.

SI

Porque representa de manera práctica el flujo de información que sostendrá la solución tecnológica propuesta.

SI

El manejo de base de datos es importante en todo desarrollo de software, y en el caso de microservicios mucho más.

Conservo mi opinión

8.- Según la metodología MCM propuesta.
¿Considera usted correcto que hacer uso del enfoque Domain Driven Design (DDD)

SI

Es correcto, El Diseño del Dominio, va a permitir definir claramente cuál es el alcance que debe de tenerse, además que con ello se puede delimitar

SI

Porque es esencial manejar un lenguaje uniforme de comunicación para reforzar el conocimiento de desarrollo, y evitar así problemas ante

SI

Teniendo que DDD considera al conocimiento del dominio para crear una aplicación, es importante contar con un lenguaje entre las personas

Conservo mi opinión

ayuda a los involucrados en el proceso de migración a manejar un mejor lenguaje ubicuo?		correctamente los costos, tiempos y riesgos del desarrollo en base al modelo propuesto.	rotación o renuncia de personal del proyecto.	que conocen el proceso y los expertos.		
9.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI retiraría/adicionaría algún elemento del modelo metodológico, para generar una metodología aplicada a una pequeña empresa?	NO	Considerar que se está proponiendo un modelo y la idea es poder evaluar sus resultados y en función a ello, poder hacer los ajustes necesarios que garanticen un mejor soporte a los objetivos propuestos como alternativa de solución.	SI	Porque aún falta mejorar actividades de verificación y validación de software (V&V), si bien no se necesita una gama amplia de pruebas V&V, al menos lo básico para una pequeña empresa.	NO	Están consideradas las etapas necesarias para un modelo de migración de una aplicación monolítica a microservicios. Conservo mi opinión
10.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI cree que la metodología MCM a través de sus 5 actividades	SI	Es correcto, es importante es poder conocer paso a paso cada una de las actividades y más aún si el proceso de abstracción es por	SI	Porque abarca muchos aspectos importantes para la migración ordenada.	SI	Está basada en un análisis de las tecnologías de la aplicación monolítica y la propuesta del uso de nuevas tecnologías hacia Conservo mi opinión

disminuye la
complejidad de
migración de
arquitectura a
microservicios?

módulos o etapas
de esta manera se
puede identificar
cada uno de los
detalles de los
elementos
intervenientes.

los microservicios.

Fuente: Elaboración propia

Tabla 20.

Respuesta del evaluador 2 en la segunda iteración del método Delphi.

Preguntas	Evaluador 1		Evaluador 2		Evaluador 3		Segunda opinión
	Rpta	Comentario	Rpta	Comentario	Rpta	Comentario	
1.- Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya analizado las características de las pequeñas empresas para poder plantear la metodología MCM de acorde a dichas características?	SI	Es correcto, porque el conocer las características va a permitir saber de primera fuente las necesidades reales que se deben de tener en cuenta y a la vez definir la delimitación del alcance de la propuesta.	SI	Porque se necesita establecer un contexto general para una propuesta ad hoc, basado en un alcance claro, tal como es una pequeña empresa.	SI	Es necesario identificar las tecnologías con las que cuentan los aplicativos monolíticos.	Conservo mi opinión
2.- Según el método generar utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya realizado un	SI	Es correcto, un punto clave y de partida es poder recopilar y analizar los estudios científicos, para que de esta manera se conozca la	SI	Porque la literatura científica es validada por académicos de alto nivel y presentan cuidado en la calidad de producción	SI	Es necesario conocer el ámbito actual de las metodologías que tratan sobre el tema de microservicios, y poder realizar una nueva propuesta.	Conservo mi opinión

análisis de la literatura científica para extraer las características, herramientas más utilizadas en la arquitectura basada en microservicios y así poder tener un concepto general para el desarrollo de la MCM propuesta?

estructura de una arquitectura orientada a microservicios y en función a ello plantear una alternativa acorde a las necesidades identificadas.

científica.

3.- Según la metodología MCM propuesta.
¿Considera usted correcto que la elaboración del modelo metodológico describe mediante las actividades a realizar, un nivel de abstracción y relación comprensible para todo el proceso de

SI

Es correcto, en el modelo planteado se define todo el conjunto de actividades y tareas que deben de ejecutarse que permitan identificar las características propias que se sigue en el proceso de la metodología en todo su ciclo de vida de desarrollo.

SI

Porque estás considerando verificación por expertos y validación con un caso de estudio.

SI

Abarca todas las etapas necesarias para descomponer y desacoplar la aplicación monolítica en una serie de microservicios.

Conservo mi opinión

la metodología
MCM propuesta?

4.- Según la metodología MCM propuesta.
¿Considera usted correcto que la elaboración de un análisis de factibilidad de migración de una arquitectura monolítica a una basada en microservicios basado en un cuestionario puede establecer un juicio correcto para validar la “factibilidad de migración” en la metodología MCM propuesta?

SI

Es Correcto, el análisis de la factibilidad va a permitir conocer si es posible hacer el proceso de migración conociendo los aspectos técnicos, económicos y operativos a considerarse que garanticen la fiabilidad del modelo a desarrollar.

NO

Porque estás considerando que el análisis lo realice un Gerente con sólo preguntas. Esto amerita que:
1. El Gerente tenga un alto nivel de experiencia en operatividad.
2. Se obtienen algunos aspectos técnicos.

SI

Es necesario conocer si es factible convertir una aplicación monolítica a microservicios.

Conservo mi opinión

5.- Según la metodología MCM propuesta.

SI

Es correcto, el objetivo de un paradigma ágil es

SI

Porque para pequeñas empresas se necesita agilidad.

SI

Como la metodología está basada en las fases

Conservo mi opinión

¿Considera usted correcto que la actividad de planeamiento de software se haga uso de una metodología ágil como Scrum, XP, con el fin de hacer mejoras continuas en el desarrollo de software?

6.- Según la metodología MCM propuesta.
¿Considera usted que para la actividad de diseño de software es correcto hacer un análisis de la tecnología actual y la tecnología a migrar a través de un cuadro comparativo para tener un concepto general de las ventajas y

SI

poder reducir los tiempos de desarrollo y tener en contacto directo a los usuarios en cada iteración que se tiene que realizar, teniendo en cuenta que un punto de partida es poder considerar las necesidades o requerimientos de usuarios.

Es correcto, la actividad de Diseño es uno de los puntos críticos, es decir, en ella se plantea el modelo de propuesta considerando la evaluación del paradigma actual y el propuesto, para ello debe de organizarse un comparativos de factores de evaluación.

SI

Porque orientará al proceso de reingeniería y en el uso de recursos sólidos ya existentes.

SI

de la ingeniería del software, es necesario el uso de una metodología ágil para que acelere el proceso.

Es necesario realizar este cuadro comparativo, sin embargo, es necesario que se cuente con expertos en el área para identificar estas ventajas y desventajas.

Conservo mi opinión

desventajas de las mismas?

7.- Según la metodología MCM propuesta.
¿Considera usted correcto hacer uso de un modelado con base en BPMN, con la finalidad de tener un mejor concepto del proceso a realizar?

SI

Es correcto, considerar que el modelar los procesos de negocios nos va poder conocer la integridad y relacionamiento de cada uno de los elementos interactuantes del sistema, a la vez que se puede generar una simulación previa del modelo y poder evaluar sus efectos y sensibilidad como parte del sistema.

SI

Porque representa de manera práctica el flujo de información que sostendrá la solución tecnológica propuesta.

SI

El manejo de base de datos es importante en todo desarrollo de software, y en el caso de microservicios mucho más.

Conservo mi opinión

8.- Según la metodología MCM propuesta.
¿Considera usted correcto que hacer uso del enfoque Domain Driven Design (DDD)

SI

Es correcto, El Diseño del Dominio, va a permitir definir claramente cuál es el alcance que debe de tenerse, además que con ello se puede delimitar

SI

Porque es esencial manejar un lenguaje uniforme de comunicación para reforzar el conocimiento de desarrollo, y evitar así problemas ante

SI

Teniendo que DDD considera al conocimiento del dominio para crear una aplicación, es importante contar con un lenguaje entre las personas

Conservo mi opinión

ayuda a los involucrados en el proceso de migración a manejar un mejor lenguaje ubicuo?		correctamente los costos, tiempos y riesgos del desarrollo en base al modelo propuesto.	rotación o renuncia de personal del proyecto.	que conocen el proceso y los expertos.		
9.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI retiraría/adicionaría algún elemento del modelo metodológico, para generar una metodología aplicada a una pequeña empresa?	NO	Considerar que se está proponiendo un modelo y la idea es poder evaluar sus resultados y en función a ello, poder hacer los ajustes necesarios que garanticen un mejor soporte a los objetivos propuestos como alternativa de solución.	SI	Porque aún falta mejorar actividades de verificación y validación de software (V&V), si bien no se necesita una gama amplia de pruebas V&V, al menos lo básico para una pequeña empresa.	NO	Están consideradas las etapas necesarias para un modelo de migración de una aplicación monolítica a microservicios. Conservo mi opinión
10.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI cree que la metodología MCM a través de sus 5 actividades	SI	Es correcto, es importante es poder conocer paso a paso cada una de las actividades y más aún si el proceso de abstracción es por	SI	Porque abarca muchos aspectos importantes para la migración ordenada.	SI	Está basada en un análisis de las tecnologías de la aplicación monolítica y la propuesta del uso de nuevas tecnologías hacia Conservo mi opinión

disminuye la
complejidad de
migración de
arquitectura a
microservicios?

módulos o etapas
de esta manera se
puede identificar
cada uno de los
detalles de los
elementos
intervenientes.

los microservicios.

Fuente: Elaboración propia

Tabla 21.

Respuesta del evaluador 3 en la segunda iteración del método Delphi.

Preguntas	Evaluador 1		Evaluador 2		Evaluador 3		Segunda opinión
	Rpta	Comentario	Rpta	Comentario	Rpta	Comentario	
1.- Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya analizado las características de las pequeñas empresas para poder plantear la metodología MCM de acorde a dichas características?	SI	Es correcto, porque el conocer las características va a permitir saber de primera fuente las necesidades reales que se deben de tener en cuenta y a la vez definir la delimitación del alcance de la propuesta.	SI	Porque se necesita establecer un contexto general para una propuesta ad hoc, basado en un alcance claro, tal como es una pequeña empresa.	SI	Es necesario identificar las tecnologías con las que cuentan los aplicativos monolíticos.	Conservo mi opinión
2.- Según el método generar utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya realizado un	SI	Es correcto, un punto clave y de partida es poder recopilar y analizar los estudios científicos, para que de esta manera se conozca la	SI	Porque la literatura científica es validada por académicos de alto nivel y presentan cuidado en la calidad de producción	SI	Es necesario conocer el ámbito actual de las metodologías que tratan sobre el tema de microservicios, y poder realizar una nueva propuesta.	Conservo mi opinión

<p>análisis de la literatura científica para extraer las características, herramientas más utilizadas en la arquitectura basada en microservicios y así poder tener un concepto general para el desarrollo de la MCM propuesta?</p>	<p>estructura de una arquitectura orientada a microservicios y en función a ello plantear una alternativa acorde a las necesidades identificadas.</p>	<p>científica.</p>		
<p>3.- Según la metodología MCM propuesta. ¿Considera usted correcto que la elaboración del modelo metodológico describe mediante las actividades a realizar, un nivel de abstracción y relación comprensible para todo el proceso de</p>	<p>Es correcto, en el modelo planteado se define todo el conjunto de actividades y tareas que deben de ejecutarse que permitan identificar las características propias que se sigue en el proceso de la metodología en todo su ciclo de vida de desarrollo.</p>	<p>Porque estás considerando verificación por expertos y validación con un caso de estudio.</p>	<p>Abarca todas las etapas necesarias para descomponer y desacoplar la aplicación monolítica en una serie de microservicios.</p>	<p>Conservo mi opinión</p>

la metodología
MCM propuesta?

4.- Según la metodología MCM propuesta.
¿Considera usted correcto que la elaboración de un análisis de factibilidad de migración de una arquitectura monolítica a una basada en microservicios basado en un cuestionario puede establecer un juicio correcto para validar la “factibilidad de migración” en la metodología MCM propuesta?

SI

Es Correcto, el análisis de la factibilidad va a permitir conocer si es posible hacer el proceso de migración conociendo los aspectos técnicos, económicos y operativos a considerarse que garanticen la fiabilidad del modelo a desarrollar.

NO

Porque estás considerando que el análisis lo realice un Gerente con sólo preguntas. Esto amerita que:
1. El Gerente tenga un alto nivel de experiencia en operatividad.
2. Se obvien algunos aspectos técnicos.

SI

Es necesario conocer si es factible convertir una aplicación monolítica a microservicios.

Conservo mi opinión

5.- Según la metodología MCM propuesta.

SI

Es correcto, el objetivo de un paradigma ágil es

SI

Porque para pequeñas empresas se necesita agilidad.

SI

Como la metodología está basada en las fases

Conservo mi opinión

<p>¿Considera usted correcto que la actividad de planeamiento de software se haga uso de una metodología ágil como Scrum, XP, con el fin de hacer mejoras continuas en el desarrollo de software?</p>	<p>poder reducir los tiempos de desarrollo y tener en contacto directo a los usuarios en cada iteración que se tiene que realizar, teniendo en cuenta que un punto de partida es poder considerar las necesidades o requerimientos de usuarios.</p>		<p>de la ingeniería del software, es necesario el uso de una metodología ágil para que acelere el proceso.</p>	
<p>6.- Según la metodología MCM propuesta. ¿Considera usted que para la actividad de diseño de software es correcto hacer un análisis de la tecnología actual y la tecnología a migrar a través de un cuadro comparativo para tener un concepto general de las ventajas y</p>	<p>Es correcto, la actividad de Diseño es uno de los puntos críticos, es decir, en ella se plantea el modelo de propuesta considerando la evaluación del paradigma actual y el propuesto, para ello debe de organizarse un comparativos de factores de evaluación.</p>	<p>Porque orientará al proceso de reingeniería y en el uso de recursos sólidos ya existentes.</p>	<p>Es necesario realizar este cuadro comparativo, sin embargo, es necesario que se cuente con expertos en el área para identificar estas ventajas y desventajas.</p>	<p>Conservo mi opinión</p>
<p>SI</p>	<p>SI</p>	<p>SI</p>	<p>SI</p>	

desventajas de las mismas?

7.- Según la metodología MCM propuesta.
¿Considera usted correcto hacer uso de un modelado con base en BPMN, con la finalidad de tener un mejor concepto del proceso a realizar?

SI

Es correcto, considerar que el modelar los procesos de negocios nos va poder conocer la integridad y relacionamiento de cada uno de los elementos interactuantes del sistema, a la vez que se puede generar una simulación previa del modelo y poder evaluar sus efectos y sensibilidad como parte del sistema.

SI

Porque representa de manera práctica el flujo de información que sostendrá la solución tecnológica propuesta.

SI

El manejo de base de datos es importante en todo desarrollo de software, y en el caso de microservicios mucho más.

El uso de un modelamiento de procesos permite conocer las fases, etapas o actividades y el flujo que se tiene, es muy importante.

8.- Según la metodología MCM propuesta.
¿Considera usted correcto que hacer uso del enfoque Domain Driven Design (DDD)

SI

Es correcto, El Diseño del Dominio, va a permitir definir claramente cuál es el alcance que debe de tenerse, además que con ello se puede delimitar

SI

Porque es esencial manejar un lenguaje uniforme de comunicación para reforzar el conocimiento de desarrollo, y evitar así problemas ante

SI

Teniendo que DDD considera al conocimiento del dominio para crear una aplicación, es importante contar con un lenguaje entre las personas

Conservo mi opinión

ayuda a los involucrados en el proceso de migración a manejar un mejor lenguaje ubicuo?		correctamente los costos, tiempos y riesgos del desarrollo en base al modelo propuesto.	rotación o renuncia de personal del proyecto.	que conocen el proceso y los expertos.		
9.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI retiraría/adicionaría algún elemento del modelo metodológico, para generar una metodología aplicada a una pequeña empresa?	NO	Considerar que se está proponiendo un modelo y la idea es poder evaluar sus resultados y en función a ello, poder hacer los ajustes necesarios que garanticen un mejor soporte a los objetivos propuestos como alternativa de solución.	SI	Porque aún falta mejorar actividades de verificación y validación de software (V&V), si bien no se necesita una gama amplia de pruebas V&V, al menos lo básico para una pequeña empresa.	NO	Están consideradas las etapas necesarias para un modelo de migración de una aplicación monolítica a microservicios. Conservo mi opinión
10.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI cree que la metodología MCM a través de sus 5 actividades	SI	Es correcto, es importante es poder conocer paso a paso cada una de las actividades y más aún si el proceso de abstracción es por	SI	Porque abarca muchos aspectos importantes para la migración ordenada.	SI	Está basada en un análisis de las tecnologías de la aplicación monolítica y la propuesta del uso de nuevas tecnologías hacia Conservo mi opinión

disminuye la
complejidad de
migración de
arquitectura a
microservicios?

módulos o etapas
de esta manera se
puede identificar
cada uno de los
detalles de los
elementos
intervenientes.

los microservicios.

Fuente: Elaboración propia

Realizando un consenso de respuestas de los evaluadores respecto a la metodología propuesta en la presente investigación se puede afirmar lo siguiente:

Sobre el método general utilizado para la elaboración de la metodología, el investigador y los evaluadores consideran correcto que conocer las características permite saber las necesidades reales de una pequeña empresa. Asimismo, identificando las tecnologías necesarias, permitirá delimitar el alcance de la propuesta.

Sobre el análisis de la literatura científica, el investigador y los evaluadores consideran correcto analizar los estudios científicos como un punto de partida para conocer el ámbito de la arquitectura orientada a microservicios y plantear una propuesta de acorde al análisis realizado.

Sobre el análisis de factibilidad de migración de arquitectura, el investigador y los evaluadores 1 y 3 están de acuerdo que dicho análisis permite conocer si es posible el proceso de migración para garantizar la fiabilidad del modelo a desarrollar. Sin embargo, el evaluador 2 observa que no considera que el análisis lo realice un gerente, porque eso conlleva a un alto nivel de experiencia en operatividad y que se obvian algunos aspectos técnicos. Pero, en la propuesta de la metodología MCM se comenta que dicha evaluación involucra al Gerente de Tecnologías de la Información, para poder tener un mejor contexto y no se obvian aspectos técnicos.

Sobre retiraría/adicionaría algún elemento del modelo metodológico, el evaluador 2 observa que se puede mejorar las actividades de verificación y validación del software. Sin embargo, el evaluador 1 y 3 consideran que se propone un modelo con las etapas bases necesarias para el proceso, permitiendo poder evaluar los resultados y en función a ello, poder hacer los ajustes necesarios que garanticen un mejor soporte a los objetivos propuestos como alternativa de solución.

Sobre si la metodología MCM a través de sus actividades permitiría disminuir la complejidad de migración a microservicios, el investigador y los evaluadores consideran correcto ya que abarca muchos aspectos importantes para la migración ordenada, está basada en un análisis de las tecnologías aplicables y permite conocer paso a paso cada una de las actividades e identificar cada uno de los detalles de los elementos intervinientes.

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

La revisión de la literatura científica permitió la comprensión de las técnicas y herramientas utilizadas en los microservicios desplegados en la nube, obteniendo 62 artículos de los cuales, mediante los criterios de inclusión, exclusión y las extracciones de datos 48 artículos comprendieron el estudio de las técnicas y herramientas utilizadas en los microservicios desplegados en la nube desde el año 2014. Según el análisis de la literatura científica se concluye que la arquitectura orientada a microservicios es un tema de auge y explorado ampliamente en los últimos años y que cada vez surgen nuevas técnicas y herramientas.

La construcción del modelo de la metodología presentó cierta dificultad al tener que conocer las fases, técnicas, enfoques, principios en los que se basan los microservicios. La revisión sistemática de la literatura y la extracción de características de las PE y la MSA. Llegando a la conclusión que la metodología propuesta permitió comprender ciertos aspectos, teniendo un enfoque ágil y estructurado con la finalidad de brindar conceptos, perspectivas y herramientas que ayudaron al proceso de migración de arquitectura, disminuyendo la complejidad al contar con una etapa de análisis de factibilidad, planificación diseño, desarrollo y por último el despliegue en la nube, logrando así la migración de una aplicación monolítica a microservicios desplegada en la nube.

El desarrollo de la metodología MCM en un caso de estudio brindó una perspectiva del negocio donde los microservicios desacoplados pueden variar según el dominio del negocio. Durante la ejecución del caso práctico hubo una dependencia del negocio sobre la orquestación y el servicio cloud, siendo Kubernetes y Digital Ocean correspondientemente las que más se adaptaban a las necesidades del negocio y no las técnicas y herramientas obtenidas del análisis de la literatura científica. Llegando a la conclusión de que las técnicas, herramientas, orquestación y el servicio cloud a usar dependen del dominio del negocio y como se adapten a sus necesidades. Asimismo, la metodología MCM aplicada al caso práctico permitió

obtener un bajo acoplamiento y una alta cohesión obteniendo los valores de 0.8 y 1.25 correspondientemente, mejorando la productividad de los desarrolladores involucrados en un 20.2% y obteniendo una “Buena” satisfacción y una complejidad “Simple”, logrando que la aplicación de la metodología MCM a un caso práctico se haya dado de manera satisfactoria.

La evaluación de la propuesta metodológica mediante un juicio de expertos permitió corroborar el planteamiento correcto de la metodología MCM a través del método Delphi. Llegando a la conclusión de que la evaluación debe de ser realizada por un grupo de expertos de manera anónima y que cuenten con experiencia en el campo de la investigación y el área de tecnologías de la información, para poder tener un consenso de opiniones acertadas sobre la metodología propuesta.

4.2. Recomendaciones

Para el desarrollo de la propuesta en un caso de estudio aplicado en una pequeña empresa se deben seguir las actividades propuestas en la metodología MCM. Sin embargo, se deben de considerar las necesidades de la empresa según su contexto, ya que ello puede tener un impacto en las tecnologías a utilizar, pero no en el proceso metodológico a seguir.

Para realizar una validación a la metodología propuesta se debe realizar una evaluación mediante el método Delphi, el cual se basa en un juicio de expertos, los cuales mediante un cuestionario analizan y evalúan la propuesta. Permitiendo tener un análisis amplio de acuerdo con sus opiniones y respuestas en las iteraciones realizadas, para así mejorar la propuesta constantemente.

REFERENCIAS

- A. Banati, E. kail, K. K. and M. K. (2018). *Authentication and authorization orchestrator for microservice-based software architectures*. 1180–1184.
- Abbott, M. L., & Fisher, M. T. (2015). *The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise* (2^a edición; Addison-Wesley, ed.). Retrieved from 0134032802
- Acevedo, C. A. J., Gómez, J. P., & Patiño, J. I. R. (2017). Methodology to transform a monolithic software into a microservice architecture. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.
- Ackoff, R. L. (1967). *The Design of Social Research* (U. of Chicago, ed.). Chicago.
- Afanasev, M. Y., Fedosov, Y. V., Krylova, A. A., & Shorokhov, S. A. (2017). An application of microservices architecture pattern to create a modular computer numerical control system. *Conference of Open Innovation Association, FRUCT, 2017-April*, 10–19. <https://doi.org/10.23919/FRUCT.2017.8071286>
- Alshuqayran, N., Ali, N., & Evans, R. (2018). Towards Micro Service Architecture Recovery: An Empirical Study. *Proceedings - 2018 IEEE 15th International Conference on Software Architecture, ICSA 2018*, 47–56. <https://doi.org/10.1109/ICSA.2018.00014>
- Amaral, M., Polo, J., Carrera, D., Mohomed, I., Unuvar, M., & Steinder, M. (2015). Performance evaluation of microservices architectures using containers. *Proceedings - 2015 IEEE 14th International Symposium on Network Computing and Applications, NCA 2015*, 27–34. <https://doi.org/10.1109/NCA.2015.49>
- Amo, F. A., Normand, L. M., & Pérez, J. S. (2005). *Introducción a la ingeniería del software: Modelos de desarrollo de programas* (Delta, ed.). España.
- Anita. (2016). What are Microservices? Retrieved from <https://www.weave.works/blog/what-are-microservices/>
- Asik, T., & Selcuk, Y. E. (2017). Policy enforcement upon software based on microservice architecture. *Proceedings - 2017 15th IEEE/ACIS International Conference on Software Engineering Research, Management and Applications, SERA 2017*, 283–287.

- <https://doi.org/10.1109/SERA.2017.7965739>
- AWS. (2019). ¿Qué es la informática en la nube? Retrieved December 12, 2019, from Amazon Web Services website: https://aws.amazon.com/es/what-is-cloud-computing/?nc1=f_cc
- Azure, M. (2019). Estilo de arquitectura de microservicios. Retrieved November 11, 2019, from <https://docs.microsoft.com/es-es/azure/architecture/guide/architecture-styles/microservices>
- Bakshi, K. (2017). Microservices-based software architecture and approaches. *IEEE Aerospace Conference Proceedings*. <https://doi.org/10.1109/AERO.2017.7943959>
- Baumeister, H., Marchesi, M., & Holcombe, M. (2005). *Extreme Programming and Agile Processes in Software Engineering* (Springer, Ed.).
- Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change* (2ª edición; Addison-Wesley, Ed.). Boston.
- Benedict, T., Kirchmer, M., Scarsig, M., Frantz, P., Saxena, R., Morris, D., & Hilty, J. (2019). *BPM CBOK Version 4.0: Guide to the Business Process Management Common Body Of Knowledge* (Independently, Ed.).
- Bucchiarone, A., Dragoni, N., Dustdar, S., Larsen, S. T., & Mazzara, M. (2018). From Monolithic to Microservices: An Experience Report from the Banking Domain. *IEEE Software*, 35(3), 50–55. <https://doi.org/10.1109/MS.2018.2141026>
- Butzin, B., Golatowski, F., & Timmermann, D. (2016). Microservices approach for the internet of things. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2016-Novem*. <https://doi.org/10.1109/ETFA.2016.7733707>
- Buzato, F. H. L., Goldman, A., & Batista, D. (2018). Efficient resources utilization by different microservices deployment models. *NCA 2018 - 2018 IEEE 17th International Symposium on Network Computing and Applications*, (i), 1–4. <https://doi.org/10.1109/NCA.2018.8548346>
- Cao, L., & Ramesh, B. (2008). Agile requirements engineering practices: An empirical study. *IEEE Software*, 25(1), 60–67. <https://doi.org/10.1109/MS.2008.1>

- Chen, R., Li, S., & Li, Z. (2018). From Monolith to Microservices: A Dataflow-Driven Approach. *Proceedings - Asia-Pacific Software Engineering Conference, APSEC, 2017-Decem*, 466–475. <https://doi.org/10.1109/APSEC.2017.53>
- Cojocar, M.-D., Oprescu, A., & Uta, A. (2019). *Attributes Assessing the Quality of Microservices Automatically Decomposed from Monolithic Applications*. (1), 84–93. <https://doi.org/10.1109/ispdc.2019.00021>
- DeFranco, J., & Laplante, P. A. (2017). Integrative Literature Review Review and Analysis of Software Development Team Communication Research. *IEEE Transactions on Professional Communication*, 00(00), 1–18. Retrieved from https://scholar.google.com/scholar?cluster=11990542706640245138&hl=en&as_sdt=0,5
- Djogic, E., Ribic, S., & Donko, D. (2018). Monolithic to microservices redesign of event driven integration platform. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, 1411–1414. <https://doi.org/10.23919/MIPRO.2018.8400254>
- Escobar, D., Cardenas, D., Amarillo, R., Castro, E., Garces, K., Parra, C., & Casallas, R. (2017). Towards the understanding and evolution of monolithic applications as microservices. *Proceedings of the 2016 42nd Latin American Computing Conference, CLEI 2016*. <https://doi.org/10.1109/CLEI.2016.7833410>
- Evans, E. (2014). *Domain-Driven Design Reference: Definitions and Pattern Summaries* (D. Ear, Ed.).
- Fan, C. Y., & Ma, S. P. (2017). Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report. *Proceedings - 2017 IEEE 6th International Conference on AI and Mobile Services, AIMS 2017*, 109–112. <https://doi.org/10.1109/AIMS.2017.23>
- Fidge, A. F. and C., Zimmermann, O., Kelly, W., & Barros, A. (2018). Migrating Enterprise Legacy Source Code to Microservices. *IEEE Software*, 63–72.
- Frank Della Rosa. (2021). *U.S. Businesses Become Future-Ready with Large-Scale Cloud Migration*.
- Gan, Y., & Delimitrou, C. (2018). The architectural implications of cloud

- microservices. *IEEE Computer Architecture Letters*, 17(2), 155–158.
<https://doi.org/10.1109/LCA.2018.2839189>
- Garai, A., Adamko, A., & Pentek, I. (2018). Reflective bio-sensory signal-processing. *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, (CogInfoCom), 185–190.
- Gartner. (2020). Gartner Forecasts Worldwide Public Cloud End-User Spending to Grow 18% in 2021. Retrieved September 10, 2021, from <https://www.gartner.com/en/newsroom/press-releases/2020-11-17-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-grow-18-percent-in-2021>
- Git. (2021). Git. Retrieved from <https://git-scm.com/>
- Gonzalez, D., RV, R., & Sharma, S. (2017). *Microservices: Building Scalable Software* (Packt, Ed.).
- Grobmann, M., & Ioannidis, C. (2019). Continuous integration of applications for onos. *Proceedings of the 2019 IEEE Conference on Network Softwarization: Unleashing the Power of Network Softwarization, NetSoft 2019*, 213–217.
<https://doi.org/10.1109/NETSOFT.2019.8806696>
- Guamán, D., Yaguachi, Lady, Cueva C, S., Jaramillo H, D., & Soto, F. (2018). Performance evaluation in the migration process from a monolithic application to microservices. *CISTI 2018 - 13th Iberian Conference on Information Systems and Technologies*.
- Hasselbring, W., & Steinacker, G. (2017). Microservice architectures for scalability, agility and reliability in e-commerce. *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceedings*, 243–246. <https://doi.org/10.1109/ICSAW.2017.11>
- Heyse, M., Bernaert, M., & Poels, G. (2012). *Keuzes Maken binnen Processen: Het Vermijden van een Russische Roulette voor de Organisaties*.
- IETF, I. E. T. F. (2015). JSON Web Token (JWT).
- Institute, P. M. (2013). *GUÍA DE LOS FUNDAMENTOS PARA LA DIRECCIÓN DE PROYECTOS* (5ta Edició). Project Management Institute.
- Jahromi, N. T., Glitho, R. H., Larabi, A., & Brunner, R. (2018). An NFV and

- microservice based architecture for on-the-fly component provisioning in content delivery networks. *CCNC 2018 - 2018 15th IEEE Annual Consumer Communications and Networking Conference, 2018-Janua*, 1–7. <https://doi.org/10.1109/CCNC.2018.8319227>
- Jambunathan, B., & Yoganathan, K. (2018). Architecture Decision on using Microservices or Serverless Functions with Containers. *Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies, ICCTCT 2018*, 1–7. <https://doi.org/10.1109/ICCTCT.2018.8551035>
- Javit. (2016). Microservicios en el mundo real. Retrieved from <https://programar.cloud/post/microservicios-en-el-mundo-real/>
- Jin, W., Liu, T., Zheng, Q., Cui, D., & Cai, Y. (2018). Functionality-Oriented Microservice Extraction Based on Execution Trace Clustering. *Proceedings - 2018 IEEE International Conference on Web Services, ICWS 2018 - Part of the 2018 IEEE World Congress on Services*, 211–218. <https://doi.org/10.1109/ICWS.2018.00034>
- Kargar, M. J., & Hanifzade, A. (2018). Automation of regression test in microservice architecture. *2018 4th International Conference on Web Research, ICWR 2018*, 133–137. <https://doi.org/10.1109/ICWR.2018.8387249>
- Kecskemeti, G., Marosi, A. C., & Kertesz, A. (2016). The ENTICE approach to decompose monolithic services into microservices. *2016 International Conference on High Performance Computing and Simulation, HPCS 2016*, 591–596. <https://doi.org/10.1109/HPCSim.2016.7568389>
- Kharbuja, R. (2016). *Designing a Business Platform using Microservices*. FAKULTÄT FÜR INFORMATIK.
- Konrad, A. (2020). El rol del software en la pandemia según el CEO de Microsoft, Sayta Nadella: “Llegó el momento.” Retrieved from <https://www.forbesargentina.com/negocios/zurich-lanza-nueva-campana-comunicacion-n8103>
- Kookarinrat, P., & Temtanapat, Y. (2016). Design and implementation of a decentralized message bus for microservices. *2016 13th International Joint Conference on Computer Science and Software Engineering, JCSSE 2016*.

<https://doi.org/10.1109/JCSSE.2016.7748869>

- Koskela, L. (2007). *Test Driven: Practical TDD and Acceptance TDD for Java Developers* (Manning, Ed.).
- Lewis, J., & Fowler, M. (2014). Microservices a definition of this new architectural term. Retrieved December 7, 2019, from Martin Fowler website: <https://martinfowler.com/articles/microservices.html>
- Lin, H., Zhao, J., Jiao, Y., Cao, J., Ouyang, H., Fang, H., & Yuan, B. (2017). *Research on Building an Innovative Electric Power Marketing Business Application System Based on Cloud Computing and Microservices Architecture Technologies*. 246–253.
- Linthicum, D. S. (2016). Practical Use of Microservices in Moving Workloads to the Cloud. *IEEE Cloud Computing*, 3(5), 6–9. <https://doi.org/10.1109/MCC.2016.114>
- Liu, X., Jiang, S., Zhao, X., & Jin, Y. (2018). A shortest-response-time assured microservices selection framework. *Proceedings - 15th IEEE International Symposium on Parallel and Distributed Processing with Applications and 16th IEEE International Conference on Ubiquitous Computing and Communications, ISPA/IUCC 2017*, 1266–1268. <https://doi.org/10.1109/ISPA/IUCC.2017.00192>
- Lucidchart. (2021). What is Business Process Modeling Notation.
- Ma, S. P., Chuang, Y., Lan, C. W., Chen, H. M., Huang, C. Y., & Li, C. Y. (2018). Scenario-Based Microservice Retrieval Using Word2Vec. *Proceedings - 2018 IEEE 15th International Conference on e-Business Engineering, ICEBE 2018, (Ddd)*, 239–244. <https://doi.org/10.1109/ICEBE.2018.00046>
- Ma, S. P., Fan, C. Y., Chuang, Y., Lee, W. T., Lee, S. J., & Hsueh, N. L. (2018). Using Service Dependency Graph to Analyze and Test Microservices. *Proceedings - International Computer Software and Applications Conference*, 2, 81–86. <https://doi.org/10.1109/COMPSAC.2018.10207>
- MakeSoft. (2016). Breve historia del Cloud Computing. Retrieved from <https://www.makesoft.es/es/breve-historia-del-cloud-computing/>
- Mehta, B. (2014). *RESTful Java Patterns and Best Practices* (PACKT, Ed.).
- Menzinsky, A., López, G., & Palacio, J. (2019). *Scrum Master* (Versión 2.; S.

- Creative, Ed.). Retrieved from https://www.scrummanager.net/files/scrum_manager.pdf
- Mersch, V. Van Der. (2016). DevOps impulsados por API: Spotlight en Docker. Retrieved November 18, 2019, from NORDIC APIS website: <https://nordicapis.com/api-driven-devops-spotlight-on-docker/>
- Microsoft. (2017). Microsoft y Soluciones Orión refuerzan su alianza para impulsar la transformación digital de empresas peruanas. Retrieved from <https://news.microsoft.com/es-xl/microsoft-soluciones-orion-refuerzan-alianza-impulsar-la-transformacion-digital-empresas-peruanas/>
- Mike, C. (2004). *User Stories Applied: For Agile Software Development* (1ra Edición; A. Wesley, Ed.).
- Moody, D. L. (2003). The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods. *EUROPEAN CONFERENCE ON INFORMATION SYSTEMS (ECIS)*.
- Murugesan, V. (2017). *Microservices Deployment Cookbook* (Packt, Ed.).
- Nadareishvili, I., Mitra, R., McLarty, M., & Amundsen, M. (2016). *Microservice Architecture Aligning Principles, Practices, and Culture* (1ª edición; O'Reilly, Ed.). Boston.
- Nakazawa, R., Ueda, T., Enoki, M., & Horii, H. (2018). Visualization Tool for Designing Microservices with the Monolith-First Approach. *Proceedings - 6th IEEE Working Conference on Software Visualization, VISSOFT 2018*, 32–42. <https://doi.org/10.1109/VISSOFT.2018.00012>
- Negocio, D. (2016). Infografía: Cloud Computing y sus beneficios para las Pymes. Retrieved from <https://destinonegocio.com/pe/gestion-pe/infografia-cloud-computing-beneficios-pymes/>
- Newman, S. (2015). *Building Microservices* (1ª edición; O'Reilly, Ed.). Boston.
- Nickoloff, J. (2016). *Docker in Action* (Manning, Ed.).
- Nil, S., Deprost, E., Bernaert, M., & Poels, G. (2012). *Van Strategie tot Procesmodellering in Kleine en Middelgrote Organisaties: Een Exploratief Onderzoek*.
- Noor, A., Jha, D. N., Mitra, K., Jayaraman, P. P., Souza, A., Ranjan, R., & Dustdar,

- S. (2019). A Framework for Monitoring Microservice-Oriented Cloud Applications in Heterogeneous Virtualization Environments. *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, 156–163. <https://doi.org/10.1109/cloud.2019.00035>
- Otero, C. (2012). *Software Engineering Design: Theory and Practice* (1ra Edició; Auerbach, Ed.).
- Pautasso, C., & Wilde, E. (2014). *REST: Advanced Research Topics and Practical Applications* (Springer, Ed.).
- Perú, C. de la R. del. (2019). LEY N° 30056 - LEY QUE MODIFICA DIVERSAS LEYES PARA FACILITAR LA INVERSIÓN, IMPULSAR EL DESARROLLO PRODUCTIVO Y EL CRECIMIENTO EMPRESARIAL. *Diario Oficial El Peruano - Normas Legales*, pp. 498461–498471.
- Perú21. (2016). Cómo hacer uso de la tecnología en la gestión de PYMES. Retrieved from <https://peru21.pe/mis-finanzas/tecnologia-gestion-pymes-211483>
- Phain, C., & Limpiyakom, Y. (2018). Scaling Network Traffic Logger with Microservice Architecture. *2018 International Conference on System Science and Engineering, ICSSE 2018*. <https://doi.org/10.1109/ICSSE.2018.8520153>
- Posadas, J. V. (2017). Application of mixed distributed software architectures for social-productive projects management in peru. *Proceedings of the 2017 IEEE 24th International Congress on Electronics, Electrical Engineering and Computing, INTERCON 2017*, 0–3. <https://doi.org/10.1109/INTERCON.2017.8079698>
- Prachitmutita, I., Aittinonmongkol, W., Pojjanasuksakul, N., Supattatham, M., & Padungweang, P. (2018). Auto-scaling microservices on IaaS under SLA with cost-effective framework. *Proceedings - 2018 10th International Conference on Advanced Computational Intelligence, ICACI 2018*, 583–588. <https://doi.org/10.1109/ICACI.2018.8377525>
- Premchand, A., & Choudhry, A. (2019). Architecture simplification at large institutions using micro services. *Proceedings of the 2018 International Conference On Communication, Computing and Internet of Things, IC3IoT 2018*, 30–35. <https://doi.org/10.1109/IC3IoT.2018.8668173>

- Pressman, R. S. (2010). *Ingeniería del software: Un enfoque práctico* (7ª edición; The McGraw-Hill Companies, Ed.). México.
- Qadri, S. S. H., & Hussaan, A. M. (2019). SOA vs MSOA Comparative Analysis. *2018 3rd International Conference on Emerging Trends in Engineering, Sciences and Technology, ICEEST 2018*, 1–8. <https://doi.org/10.1109/ICEEST.2018.8643315>
- Quintos, M. G. (2016). ¿Cómo está la industria de Software en Perú? Retrieved November 10, 2019, from <http://www.timov.la/article/como-esta-la-industria-de-software-en-peru>
- RedHat. (2019). ¿Qué son los microservicios? Retrieved November 10, 2019, from <https://www.redhat.com/es/topics/microservices>
- Richardson, C. (2018a). Microservice Architecture. Retrieved from <https://microservices.io/patterns/microservices.html>
- Richardson, C. (2018b). Monolithic Architecture. Retrieved from <http://microservices.io/patterns/monolithic.html>
- Riol, E. (2017). BDD + TDD para descubrir el diseño de tu código. Retrieved November 18, 2019, from <https://enmilocalfunciona.io/bdd-tdd-para-descubrir-el-diseno-de-tu-codigo/>
- Sadaphal, V., & Natu, M. (2016). *Workload Characterization for*. (c), 84–91.
- Salah, T., Zemerly, M. J., Yeun, C. Y., Al-Qutayri, M., & Al-Hammadi, Y. (2017). The evolution of distributed systems towards microservices architecture. *2016 11th International Conference for Internet Technology and Secured Transactions, ICITST 2016*, 318–325. <https://doi.org/10.1109/ICITST.2016.7856721>
- Samal, C., Dubey, A., & Ratliff, L. (2018). Mobilytics-an extensible, modular and resilient mobility platform. *Proceedings - 2018 IEEE International Conference on Smart Computing, SMARTCOMP 2018*, 356–361. <https://doi.org/10.1109/SMARTCOMP.2018.00029>
- Sampieri, R. H., Collado, C. F., & Lucio, P. B. (2006). *Metodología de la Investigación* (4ª edición; McGraw-Hili, Ed.). Mexico.
- Sarkar, S., Vashi, G., & Abdulla, P. P. (2018). Towards Transforming an Industrial Automation System from Monolithic to Microservices. *IEEE International*






- Conference on Emerging Technologies and Factory Automation, ETFA, 2018-Septe*, 1256–1259. <https://doi.org/10.1109/ETF.A.2018.8502567>
- Schwaber, K. (1997). SCRUM Development Process. *Business Object Design and Implementation*, (April 1987), 117–134. https://doi.org/10.1007/978-1-4471-0947-1_11
- Singh, C., Gaba, N. S., Kaur, M., & Kaur, B. (2019). Comparison of Different CI/CD Tools Integrated with Cloud Platform. *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 7–12. <https://doi.org/10.1109/confluence.2019.8776985>
- Singh, V., & Peddoju, S. K. (2017). Container-based microservice architecture for cloud applications. *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017, 2017-Janua*, 847–852. <https://doi.org/10.1109/CCAA.2017.8229914>
- Smart, J. F. (2014). *BDD in Action: Behavior-Driven Development for the whole software lifecycle* (Manning, Ed.).
- Sommerville, I. (2011). *Ingeniería de Software* (9ª edición; Pearson Educación, Ed.). México.
- Sotomayor, J. P., Allala, S. C., Alt, P., Phillips, J., King, T. M., & Clarke, P. J. (2019). Comparison of Runtime Testing Tools for Microservices. *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, 2, 356–361. <https://doi.org/10.1109/compsac.2019.10232>
- Stephens, M., & Rosenberg, D. (2010). *Design Driven Testing: Test Smarter, Not Harder* (1ª Edición; Apress, Ed.).
- Straub, S., & Ben, C. (2014). *Pro Git* (2da Edición; Apress, Ed.).
- Stubbs, J., Moreira, W., & Dooley, R. (2015). Distributed Systems of Microservices Using Docker and Serfnode. *Proceedings - 7th International Workshop on Science Gateways, IWSG 2015*, 34–39. <https://doi.org/10.1109/IWSG.2015.16>
- Sun, L., Li, Y., & Memon, R. A. (2017). *Sun_2017_An Open IoT Framework Based on Microservices Architecture_China Commun.* 154–162.
- Suresh Kumar, S., & Mallikarjuna Shastry, P. M. (2018). Database-per-service for e-learning system with micro-service architecture. *Proceedings of the 2017*

- International Conference On Smart Technology for Smart Nation, SmartTechCon* 2017, 705–708.
<https://doi.org/10.1109/SmartTechCon.2017.8358462>
- Synergy Research Group. (2020). *COVID-19 Fails to Dent Aggressive Growth in Cloud Spending; Half of the Q1 Market Belongs to Amazon & Microsoft*. Retrieved from <https://www.srgresearch.com/articles/covid-19-fails-dent-aggressive-growth-cloud-spending-half-q1-market-belongs-amazon-microsoft>
- Takeda, T., Takahashi, M., Yumoto, T., Masuda, S., Matsuodani, T., & Tsuda, K. (2019). Applying change impact analysis test to migration test case extraction based on IDAU and graph analysis techniques. *Proceedings - 2019 IEEE 12th International Conference on Software Testing, Verification and Validation Workshops, ICSTW* 2019, 131–139.
<https://doi.org/10.1109/ICSTW.2019.00041>
- Torkura, K. A., Sukmana, M. I. H., Cheng, F., & Meinel, C. (2017). Leveraging Cloud Native Design Patterns for Security-as-a-Service Applications. *Proceedings - 2nd IEEE International Conference on Smart Cloud, SmartCloud 2017*, 90–97.
<https://doi.org/10.1109/SmartCloud.2017.21>
- Vaughn, V. (2013). *Implementing Domain-Driven Design* (1ra Edició; A. Wesley, Ed.).
- Vaughn, V. (2016). *Domain-Driven Design Distilled* (P. Educación, Ed.).
- Vernon, V. (2016). *Domain-Driven Design Distilled* (Addison-Wesley, Ed.). Boston.
- Villamizar, M., Garcés, O., Castro, H., Verano, M., Salamanca, L., Casallas, R., & Gil, S. (2015). Evaluating the Monolithic and the Microservice Architecture Pattern to Deploy Web Applications in the Cloud. *Manager*.
<https://doi.org/10.1109/ColumbianCC.2015.7333476>
- Villamizar, M., Garces, O., Ochoa, L., Castro, H., Salamanca, L., Verano, M., ... Lang, M. (2016). Infrastructure Cost Comparison of Running Web Applications in the Cloud Using AWS Lambda and Monolithic and Microservice Architectures. *Proceedings - 2016 16th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2016*, 179–182.
<https://doi.org/10.1109/CCGrid.2016.37>
- Wang, F. J., & Fahmi, F. (2018). Constructing a service software with microservices.

- Proceedings - 2018 IEEE World Congress on Services, SERVICES 2018*, 33–34. <https://doi.org/10.1109/SERVICES.2018.00035>
- Warke, A., Mohamed, M., Engel, R., Ludwig, H., Sawdon, W., & Liu, L. (2018). Storage service orchestration with container elasticity. *Proceedings - 4th IEEE International Conference on Collaboration and Internet Computing, CIC 2018*, 283–292. <https://doi.org/10.1109/CIC.2018.00046>
- Waseem, M., & Liang, P. (2018). Microservices architecture in DevOps. *Proceedings - 2017 24th Asia-Pacific Software Engineering Conference Workshops, APSECW 2017, 2018-Janua(61472286)*, 13–14. <https://doi.org/10.1109/APSECW.2017.18>
- World, E. F. (2016). *The Global Information Technology Report 2016 Innovating in the Digital Economy*. Ginebra, Suiza: World Economic Forum.
- Zheng, X., Jiang, J., Zhang, Y., Deng, Y., Fu, M., Zheng, T., & Liu, X. (2018). SmartVM: A multi-layer microservice-based platform for deploying SaaS. *Proceedings - 15th IEEE International Symposium on Parallel and Distributed Processing with Applications and 16th IEEE International Conference on Ubiquitous Computing and Communications, ISPA/IUCC 2017*, 470–474. <https://doi.org/10.1109/ISPA/IUCC.2017.00077>

ANEXOS.

Anexo 1. Resolución de aprobación del proyecto de investigación.

 UNIVERSIDAD SEÑOR DE SIPÁN	<p style="text-align: center;">FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO RESOLUCIÓN N°1000-A-2021/FIAU-USS</p> <p style="text-align: center;">Pimentel, 11 de noviembre de 2021</p> <p>VISTO: El oficio N° 0359-2021/FIAU-IS-USS de fecha 14 de octubre de 2021, de la Dirección de Escuela profesional de INGENIERÍA DE SISTEMAS con el que remite el Acta de reunión N°0610-2021 del Comité de investigación de la referida Escuela profesional, acerca de la Tesis presentada por estudiantes del Programa de estudios de INGENIERÍA DE SISTEMAS, y;</p> <p>CONSIDERANDO:</p> <p>Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";</p> <p>Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y <u>tesis</u> son <u>aprobados por el Comité de Investigación</u> y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El <u>periodo de vigencia de los mismos será de dos años</u>, a partir de su aprobación. En caso un tema perdiera vigencia, <u>el Comité de Investigación evaluará la ampliación de la misma</u>.</p> <p>Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; <u>es individual o en pares para obtener un título profesional</u>. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C."</p> <p>Que, mediante documentos de vistos, el Comité de investigación de la referida Escuela profesional acordó aprobar la ampliación de la vigencia de las tesis que se detallan en el Acta de reunión N°0610-2021, a cargo de estudiantes del Programa de estudios INGENIERÍA DE SISTEMAS, hasta el 6 de octubre de 2023.</p> <p>Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;</p> <p>SE RESUELVE:</p> <p>ARTÍCULO ÚNICO: AMPLIAR VIGENCIA, de la Tesis a cargo de los estudiantes del Programa de estudios de INGENIERÍA DE SISTEMAS que se detallan en el anexo de la presente Resolución, hasta el 6 de octubre de 2023.</p> <p>REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE</p> <div style="display: flex; justify-content: space-around; align-items: center;"><div style="text-align: center;">  Mg. Víctor Alexei Tuesta Montaña Decano (t) / Facultad De Ingeniería, Arquitectura Y Urbanismo UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.</div><div style="text-align: center;">  MBA. María Noelia Slater Rivera Secretaría Académica / Facultad de Ingeniería, Arquitectura y Urbanismo UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.</div></div> <p style="text-align: center;">Cc: Interesado, Archivo</p>
---	--

**FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°1000-A-2021/FIAU-USS**

Pimentel, 11 de noviembre de 2021

ANEXO

N°	APELLIDOS Y NOMBRES	TEMA DE TESIS	FECHA RESOLUCIÓN DE APROBACIÓN/MODIFICACIÓN TEMA DE TESIS/AMPLIACIÓN DE VIGENCIA
1	DÍAZ CARRASCO NATIVIDAD ALEJANDRO	EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS	13-05-2016
2	MENDOZA LINARES JERSSON GERMAN	METODOLOGÍA DE CONVERSIÓN DE APLICACIONES MONOLÍTICAS A MICROSERVICIOS DESPLEGABLE EN LA NUBE PARA PEQUEÑAS EMPRESAS	22-07-2019
3	GONZALEZ FLORES PAUL GUSTAVO	ANÁLISIS COMPARATIVO DE ALGORITMOS DE CLASIFICACIÓN PARA DIAGNOSTICAR TIPOS DE LEUCEMIA INFANTIL	17-11-2020
4	SOPLOPUCO MONJA BRAYAN ALONSO	COMPARACIÓN DE TÉCNICAS CONSTRUCCIÓN DE PROTOTIPOS Y REVISIÓN DE REQUERIMIENTOS PARA REALIZAR UNA CORRECTA VALIDACIÓN DE REQUERIMIENTOS DE SOFTWARE	Año 2018
5	MONTENEGRO GUERRERO VICTOR AGUSTIN	ANALISIS COMPARATIVO DE ALGORITMOS DE MACHINE LEARNING PARA DETECCION DE MALWARE EN APLICACIONES ANDROID	Año 2019

Anexo 2. Carta de aceptación de la institución para la recolección de datos.

CARTA DE ACEPTACIÓN

Chiclayo, 01 de septiembre del 2021

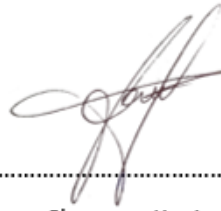
Universidad Señor de Sipán

Por medio de la presente, HTEC E.I.R.L identificado con el R.U.C 20604657785, nos permitimos notificar la ACEPTACIÓN del proyecto "GMO PLUS" en el cual se aplicará la "METODOLOGÍA DE CONVERSIÓN DE APLICACIONES MONOLÍTICAS A MICROSERVICIOS DESPLEGABLE EN LA NUBE PARA PEQUEÑAS EMPRESAS" a llevarse a cabo por Jersson German Mendoza Linares, identificado con el documento de identidad 73932770, alumno de la carrera profesional de Ingeniería de Sistemas de la Universidad Señor de Sipán.

Como se indicó en la presentación del proyecto "GMO PLUS", Jersson German Mendoza Linares estará a cargo de las características del proyecto a desarrollar. Asimismo, se solicita total confidencialidad en el código, UI/UX de los aplicativos a analizar y desarrollar.

Esperamos que la culminación de este proyecto se lleve a cabo bajo las condiciones y características estipuladas.

Atentamente.



.....
Vilcherrez Chavarry Kevin Leandro
Gerente General – HTEC E.I.R.L

Anexo 3. Instrumentos de recolección de datos, con su respectiva validación de instrumentos.

Anexo 3.1. Resumen de la propuesta MCM.

METODOLOGÍA DE CONVERSIÓN DE APLICACIONES MONOLÍTICAS A MICROSERVICIOS DESPLEGABLE EN LA NUBE PARA PEQUEÑAS EMPRESAS

La propuesta de la metodología MCM se enfoca en la conversión de una aplicación monolítica a microservicios. Las plataformas sobre las que se desarrolla software están en constante mejora y desarrollo, un estudio realizado por Destino Negocio en colaboración con Movistar nos brinda la perspectiva del Cloud Computing detallando los beneficios, tales como la optimización de personal, velocidad en los negocios, eficiencia en procesos, confiabilidad en operaciones y sostenibilidad. Estos beneficios son muy importantes, siendo una de las principales razones para que las pequeñas y medianas empresas (Pymes) migren paulatinamente hacia la nube, a esto se suma las copias de seguridad, ahorro en infraestructura y accesos remotos, a sí mismo el 55% de empresas ven como prioridad la nube para el desarrollo y despliegue de sus aplicaciones y/o servicios. Las Pymes hacen uso de las Tecnologías de la Información (TI) para poder obtener potenciar su crecimiento. Además de mejorar sus procesos para poder organizar y tener un mejor control de sus datos, además de reducir sus costos entre un 15% y 20% el tiempo de los procesos operacionales también disminuyen, esto sucede solo si el software que utiliza la empresa cumple con sus necesidades. Asimismo, los datos reflejados por Perú 21 nos muestran que en temas de porcentaje las compañías que hacen uso de un software de gestión son representadas por el 55%. Por otra parte, las empresas que desarrollan software a medida están representadas por el 21%.

La pandemia del covid-19 generó un cambio en la industria tecnológica y de software. Según una investigación realizada por el Synergy Research Group, indica que durante los primeros 4 meses de la pandemia del 2020, los servicios tecnológicos de empresas como Amazon Web Services (AWS) teniendo un crecimiento del 32%, Microsoft Azure alcanzando el 18% y Google Cloud quien están superando sustancialmente el crecimiento general del mercado y están ganando cuota de mercado, reflejando un aumento del 37% en la demanda de servicios en la nube. Los servicios públicos de IaaS y PaaS representaron la mayor parte del mercado y creció un 39% en el primer trimestre. Una encuesta realizada por IDC indicó que, de 200 líderes de tecnologías de la información, el

27,5% confirmó que la migración a la nube era esencial para el sector empresarial. Sin embargo, el 17% de las empresas ralentizaron sus planes de digitalización debido a la pandemia. Asimismo, el 66% admitió que debido a la pandemia se aceleró la implementación a la nube. De tal manera, que el 85% de líderes admitieron que el cambio permitió a las compañías una mejor posición frente a futuras crisis o problemáticas al proporcionar ahorros en los costos y mayor eficiencia. El instituto de investigación Gartner pronosticó un crecimiento mundial del 18,4% para el año 2021, la infraestructura PaaS tendrá un crecimiento del 26,6. Los servicios en la nube cada vez están siendo más usados por las empresas y diversos sectores en la industria, permitiendo desarrollar aplicaciones de alto rendimiento y escalables.

Según Satya Nadella, durante la pandemia el cloud computing ha tenido una migración significativa. Asimismo, en el ámbito del desarrollo de software GitHub ha tenido un aumento del 25% de colaboraciones durante los primeros meses de la pandemia. De la misma forma, las soluciones tecnológicas a través del desarrollo de software se han incrementado y tenido mayor importancia en la industria, resolviendo problemas de servicios, conectividad y disponibilidad. No obstante, las empresas encargadas de brindar estos servicios de software han tenido la necesidad de mejorar su escalabilidad, disponibilidad y tolerancia a fallos en sus sistemas brindados, dado que la demanda sobre este tipo de sistemas se ha visto en aumento. El software desarrollado y desplegado en la nube comúnmente se desarrolla de una manera monolítica lidiando con la capacidad del servidor donde se está ejecutando, además del exceso de usuarios y falta de modularidad, para ello la alternativa utilizar microservicios, pensemos en la aplicación como bloques de construcción, esto nos permitirá mejor escalabilidad, disponibilidad, versatilidad, mejoras continuas. Sin embargo, el proceso de conversión genera cierta complejidad por lo que una metodología de conversión ayudaría a tener un enfoque de cómo realizar la conversión.

Por consecuente, se pretende brindar una propuesta metodológica para de conversión de aplicaciones monolíticas a microservicios desplegable en la nube para pequeñas empresas (MCM), por ello se planteó el siguiente modelo general:

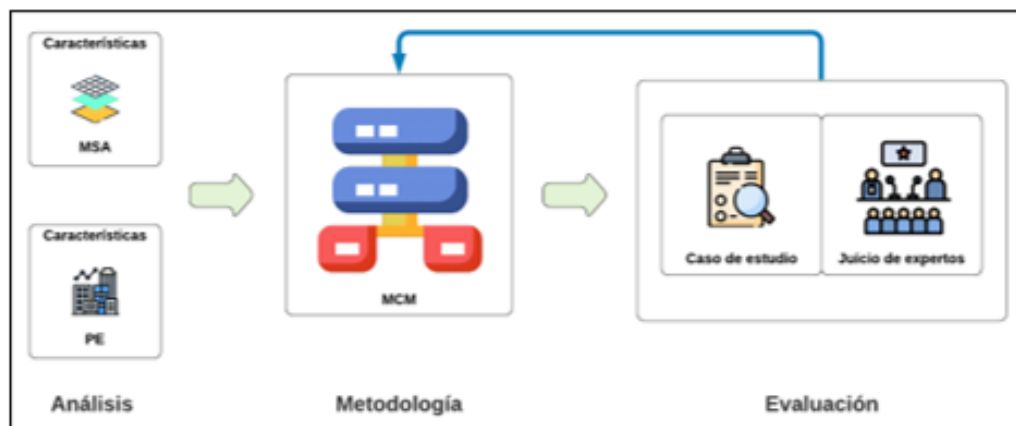


Figura 1: Método general de la investigación.

La metodología MCM, considera las características de la arquitectura basada en microservicios (MSA) y las pequeñas empresas (PE). Con base a ello se realiza la elaboración de la metodología reflejando el proceso a seguir para la implementación en PE. La metodología se basa en 5 actividades "Análisis de factibilidad de migración de arquitectura de software", "Planeamiento de Software", "Diseño de Software", "Desarrollo de Software", "Despliegue de Software"

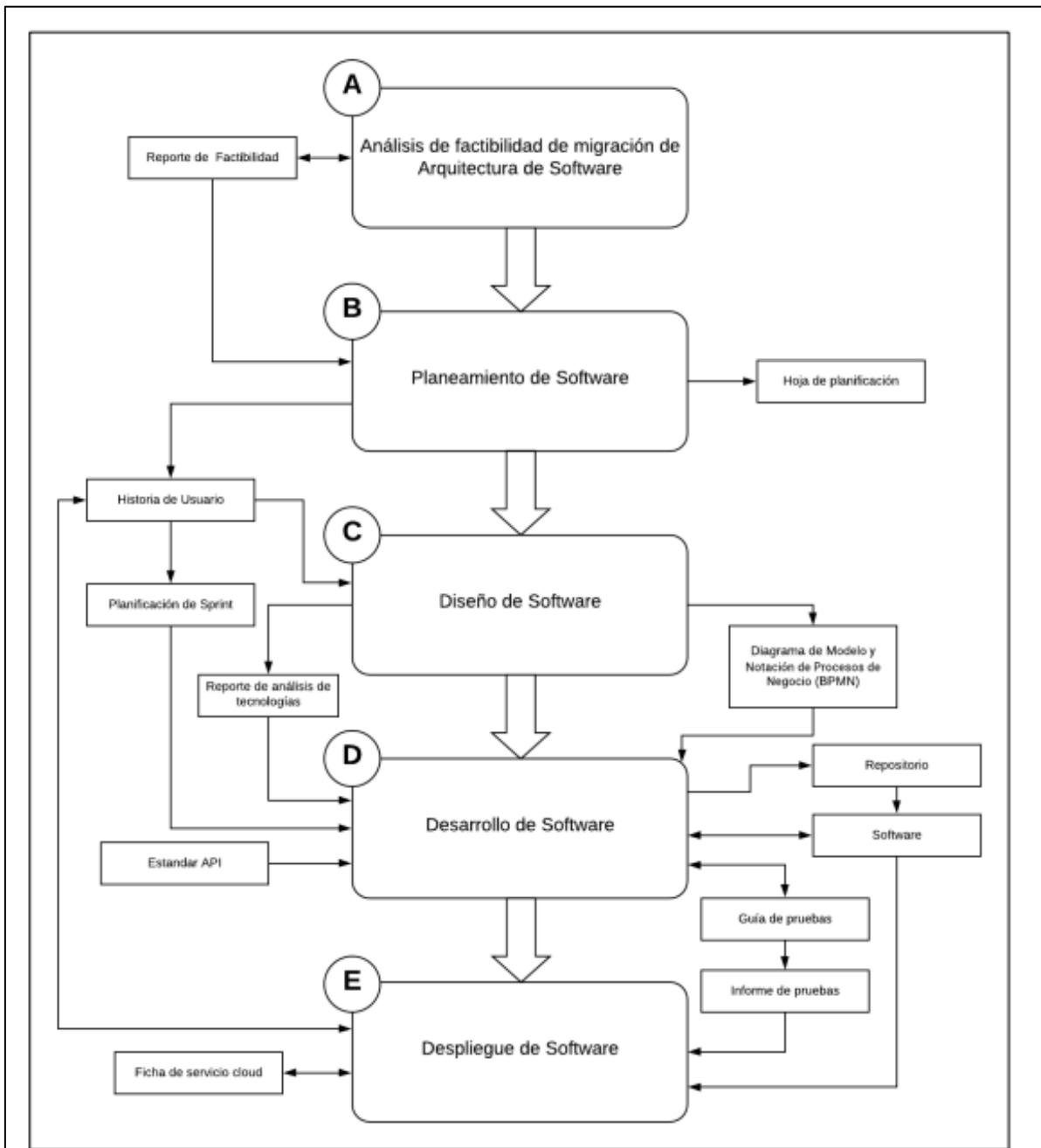


Figura 2: Metodología de conversión de aplicaciones monolíticas a microservicios desplegable en la nube para pequeñas empresas (MCM).

A.- El análisis de factibilidad de migración de arquitectura de software, se basa en ayudar a definir mediante preguntas el enfoque sobre el software que se usa actualmente y si este cumple con los objetivos de tecnologías de la información del negocio, luego de tener un enfoque claro se establece una tabla donde se califica a través de una valoración si es factible realizar la migración a la arquitectura de microservicios.

Reporte de Factibilidad. – El reporte factibilidad permite obtener los detalles necesarios para la toma de decisión sobre la migración de arquitectura de software. Por ello es necesario saber sobre la arquitectura en la cual se encuentra el software. Las arquitecturas tradicionales, monolíticas constan de un empaquetado donde se encuentran unidos la vista, lógica de negocio, acceso a datos y base de datos. La arquitectura de software actual debe al menos soportar los procesos de negocio, esto quiere decir que se debe de considerar los riesgos e inconvenientes que se están generando por esta arquitectura, para así poder evaluar cómo podría mejorar el rendimiento y disponibilidad del software al cambiar de arquitectura. Sin embargo, para ello se debe de evaluar la arquitectura actual, dicha evaluación se debe realizar al Gerente de Tecnologías de la Información

B.- El planeamiento del software, permite definir una hoja de planeamiento donde se describe el proyecto de migración que se va a realizar y el por qué. Además de tener una base en la metodología ágil a través de la historia de usuario donde se especifican los requerimientos del sistema, también se cuenta con un planeamiento del sprint a realizar y un seguimiento diario el cual permite detallar los inconvenientes que suceden diariamente durante el desarrollo del software.

Hoja de Planificación. - Luego de haber aplicado el análisis de factibilidad de migración de arquitectura de software y haber evaluado sus resultados, siendo que el resultado de la evaluación indica que es factible se llevará a cabo el planeamiento, para lo cual nos basaremos en el modelo del Acta de Constitución de la guía de PMBOK, lo cual es un proceso para definir la existencia de un proyecto, sirviendo como directriz para tener una concepción clara, sencilla y formal de lo que se va a

realizar, además brindar información vital y dar la autoridad para la asignación de los recursos necesarios para llevar a cabo las actividades.

Historia de usuario. - Es la encargada de especificar que funcionalidades son importantes para el usuario y el desarrollo, además de especificar el tiempo estimado, prioridad, validaciones, entre otros, con el fin de ayudar a conceptualizar lo que se requiere. La historia de usuario se debe ser implementada y descrita en un lenguaje común para el usuario, debe de estar estructurada siguiendo una narrativa del universo de discurso, cada historia debe de ser limitada ya que en el momento de la implementación se pueden surgir cambios en el negocio o esquema lo cual es bienvenido en el desarrollo ágil de software, para ello deben de ser acompañadas de conversaciones y tener la definición de las validaciones asociadas, las validaciones o criterios de validación permiten ratificar si se han recogido de manera correcta los requisitos, estas pruebas pueden ser adecuadas con TDD, BDD o ambos en conjunto, finalmente lo antes descrito servirá para comprobar si la historia de usuario ha sido completada totalmente.

Planificación de Sprint. - El Sprint es un ciclo o iteración que permite tener un ritmo y tiempo prefijado, esperando una duración aproximada de 4 semanas. Cada Sprint debe de generar un entregable del producto. La planificación del sprint se debe de realizar o tener como guía el siguiente formato.

C.- El diseño de software, permite definir las tecnologías a utilizar durante la migración realizándose a través de un cuadro de comparación de tecnologías. Para la conceptualización y el enfoque de los microservicios se usa Domain Driver Design, brindando la manera de tener el enfoque del modelo del dominio, el cual junto con diagramas de modelos y notación de procesos de negocio (BPMN) se podrá tener el conocimiento del proceso de negocio extrayendo los subdominios como un microservicio.

Análisis de Tecnologías. - Cuando se realiza la migración de arquitectura a microservicios se puede reevaluar el uso de las tecnologías que hace uso el software o las que se necesitan para lograr un mejor desarrollo, en base a que el concepto de microservicios es versátil en cuanto a la integración de tecnologías, por lo que se pueden comparar las tecnologías con el fin de mejorar el software, asimismo se puede actualizar las tecnologías a utilizar por lo que un cuadro comparativo brinda un mayor enfoque y contexto, para que en base a los resultados de la evaluación de tecnologías se brinde las acciones correspondientes.

Diagrama de Modelo y Notación de Procesos de Negocio (BPMN). - Es una notación gráfica estándar popular para el modelamiento de procesos de extremo a extremo que proporciona coherencia y garantiza su comprensión entre las partes interesadas y entre las organizaciones. Los diagramas de BPMN utilizan símbolos básicos para visualizar los procesos de negocio, cada uno de los cuales se puede categorizar en uno de cuatro grupos: objetos de flujo (eventos, actividades y puertas de enlace), objetos de conexión (flujos de secuencia, flujos de mensajes y asociaciones), diagramas de flujo (cuadros y líneas) y artefactos (objetos de datos, grupos y anotaciones).

D.- El desarrollo del software, permite definir el uso de un Sistema de control de versiones para el mejor control del software comprendiendo los estándares de desarrollo a través de una ficha de APIS. Además de centrarse en los casos de prueba a través de una guía de pruebas, la cual se refleja en un informe de las pruebas realizadas al software. La propuesta metodológica, permite hacer iteraciones y mejoras continuas durante el proceso de desarrollo de software.

Repositorio. - El desarrollo de software ha estado dividido en versiones y debe de estar almacenado en un solo lugar. El VCS gestiona la historia del proyecto, ofreciendo la posibilidad de recuperar una versión anterior o permitiendo el movimiento entre las versiones con la finalidad de modificar, mejorar, revisar el software desarrollado. Git es un sistema de control de versiones de código libre, permite gestionar los cambios

realizados durante el desarrollo, además de brindar información de las personas que realizaron las modificaciones y el motivo.

Software. – Es el código desarrollado que permiten que la computadora pueda desempeñar tareas inteligentes, con instrucciones y datos a través de diferentes tipos de lenguajes de programación.

Estándar API. - El desarrollo de API's es muy común para la comunicación entre microservicios, por ello es óptimo contar con un estándar de representación donde se pueda llegar a un entendimiento entre todo el equipo

Módulo. – Nombre del módulo al que pertenece

Funcionalidad. – Describe la funcionalidad que realiza la API

API. – Describe la URI a consumir

Método. – Describe el verbo a utilizar (POST, GET, PUT, DELETE)

Entrada. – Describe el formato y los datos que necesita la API para realizar una lógica.

Salida. – Describe el formato y los datos ya procesados.

Ficha de API'S					
Modulo	Funcionalidad	API	Metodo	Entrada	Salida

Figura 3. Fichas de Apis. Fuente: Elaboración propia.

Test Driven Development / Desarrollo guiado por pruebas. - Con TDD, las pruebas se escriben verificando que fallen, se implementa el código haciendo que las pruebas pases satisfactoriamente y luego se refactoriza el código mejorando la estructura y la lógica utilizada. El propósito de TDD es lograr un código que funcione, sea fácil de entender y limpio.

Para el correcto desarrollo de los test a continuación se presenta una guía de pruebas y un informe de errores.

Guías de prueba

Guía de pruebas						
Número y Nombre de Historia de usuario:						
Personas involucradas:						
Ambiente de pruebas:			Cronograma:			
Observaciones:						
Casos de Prueba						
ID	Módulo	Descripción	Ruta	Resultado esperado	Resultado de desarrollo	Resultado de testing

Figura 4. Guía de pruebas. Fuente: Elaboración propia.

Informe de pruebas

Informe de Pruebas		
Número y Nombre de Historia de usuario:		
Personas involucradas:		Fecha:
Descripción:		
Pruebas	Número	Porcentaje
Casos de pruebas ejecutadas		
Casos de pruebas con éxito		
Casos de pruebas con error		
Casos de pruebas enviadas para la corrección		

Figura 5. Informe de pruebas. Fuente: Elaboración propia.

E.- El despliegue de software, se realiza a través de un servicio cloud el cual debe contar con ciertos servicios que permiten el adecuado despliegue de los microservicios para ello se brinda una ficha cloud donde mediante un check list se evalúa los servicios cloud. Además, se cuenta con una ficha de microservicios donde se detalla el microservicio desplegado, en qué servicio cloud, el costo, para así tener un informe detallado de los microservicios que se encuentran desarrollados.

Ficha de servicios cloud. - Los servicios cloud son recursos de tecnologías de la información que funcionan bajo demanda, obteniendo servidores, bases de datos y almacenamiento en una arquitectura soportada por un proveedor de servicios. El uso del cloud cuenta con una amplia gama de usos para diferentes organizaciones para ofrecer distintos servicios. El cloud ofrece agilidad, elasticidad, ahorro en costes de infraestructura, implementación a nivel mundial.

Para poder realizar un contrato con un servicio cloud para alojar nuestro software basado en microservicios se deben de tener en cuenta ciertos servicios, por lo que a través de la siguiente ficha se puede realizar la evaluación.

Ficha de Servicio Cloud		
Nombre del servicio:		
Tipo de Servicio:	Costo de servicio	
Descripción		
¿Cuenta con servicios ?	SI	NO
Load Balancer		
Container		
Orchestrator		
Continuous Deployment		

Figura 6. Ficha de servicio cloud. Fuente: Elaboración propia.

Anexo 3.2. Perfiles de los evaluadores del método Delphi.

Experto 01

	<p>ING. FUENTES ADRIANZÉN DENNY JOHN Informático y de Sistemas. Egresado de la Escuela Profesional de Ingeniería de Computación y Sistemas de la Universidad Privada Antenor Orrego. Docente Universitario Escuela Profesional de Ingeniería en Computación e Informática Facultad de Ciencias Físicas y Matemáticas Universidad Nacional Pedro Ruiz Gallo</p> <p>EXPERIENCIA DOCENTE UNIVERSITARIA: Universidad de San Martín de Porres, Universidad César Vallejo, Universidad Nacional Pedro Ruiz Gallo, Universidad Señor de Sipán, Universidad Alas Peruanas, Universidad Particular de Chiclayo</p> <p>EXPERIENCIA LABORAL: GLOBAL BUSINESS AMERICAN S.A.C. IMPORT & EXPORT SAN IGNACIO S.A.C. LUCKY CAR IMPORT SRLTDA C E O P BUSINESS COMP S.R.L. TURISMO ADRIANZEN S.R.LTDA.</p> <p>PROYECTOS DE INVESTIGACIÓN: Plataformas Colaborativas para Apoyar la Labor Académica y de Investigación en docentes de la Universidad Nacional Pedro Ruiz Gallo - 2018.</p> <p>Modelo de Gestión de Procesos para la Mejora del Desarrollo de Software ERP - PYMES en la Empresa Red Chiroque EIRL.</p>
--	--

Experto 02



MG. CACHAY MACO JUNIOR EUGENIO
Informático y de Sistemas. Maestro en
Ingeniería de Sistemas, Especialidad:
con Mención en Gerencia de Tecnologías
De La Información y Gestión del Software
Profesional en Tecnologías de la
Información especializado en Auditorías
de Sistemas de la Información y
Desarrollo de Software. Investigaciones
en Gobierno de Tecnologías de la
Información y Generación de Valor a
través de Inversiones en proyectos de
Tecnologías de la Información.

EXPERIENCIA DOCENTE
UNIVERSITARIA: UNIVERSIDAD
NACIONAL PEDRO RUIZ GALLO
UNIVERSIDAD SEÑOR DE SIPAN
S.A.C.

EXPERIENCIA LABORAL: AUDIT AND
CONTROL OF INFORMATION
SYSTEMS S.A.C. SIEMPRESOFT
E.I.R.L.

Experto 03

	<p>ING. CARLOS ALBERTO VALDIVIA SALAZAR. Ingeniero en Computación e Informática. Maestro en Ingeniería de Sistemas, Profesional en Tecnologías de la Información especializado en Ingeniería de Software, Sistemas de la Información y Desarrollo de Software. Especialización en Transformación Digital.</p> <p>EXPERIENCIA DOCENTE UNIVERSITARIA:</p> <ul style="list-style-type: none">- Universidad Nacional Pedro Ruiz Gallo.- Universidad Señor de Sipán S.A.C. <p>Experiencia en asesoría de tesis de pregrado y posgrado.</p> <p>PUBLICACIONES CIENTÍFICAS</p> <ul style="list-style-type: none">- Digital Transformation Model for the Reactivation of the Tourism Sector in the COVID-19 Environment of the Lambayeque Region (en SCOPUS - 2021).- El Geomarketing como estrategia para el posicionamiento de una óptica en Chiclayo 2018 (En ALICIA – 2019)- Marketing de contenidos como estrategia para mejorar la decisión de compra de un Restaurante De Puerto Eten (en ALICIA – 2018)- Sistema Experto para fomentar el turismo en la región Lambayeque – Perú (en ALICIA - 2017).-
--	---

Anexo 3.3. Cuestionario del método Delphi.

CUESTIONARIO PARA LA MEDICIÓN DE LA VIABILIDAD DE LA PROPUESTA DE METODOLOGÍA DE CONVERSIÓN DE APLICACIONES MONOLÍTICAS A MICROSERVICIOS DESPLEGABLE EN LA NUBE PARA PEQUEÑAS EMPRESAS ELABORADA POR JERSSON GERMAN MENDOZA LINARES

Consideraciones:

- Las respuestas se mantendrán en el anonimato bajo el conocimiento del investigador para que la aplicación del método Delphi sea correcto.
- Complete con una "X" en alguna de las alternativas ("A" o "B"), luego se solicita redactar su respuesta a la pregunta "¿Por qué?" en las casillas de color celeste

PREGUNTA 1

Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya analizado las características de las pequeñas empresas para poder plantear la metodología MCM de acorde a dichas características?

A)	SI	<input type="checkbox"/>
B)	NO	

¿Por qué?

PREGUNTA 2

Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya realizado un análisis de la literatura científica para extraer las características, herramientas más utilizadas en la arquitectura basada en microservicios y así poder tener un concepto general para el desarrollo de la MCM propuesta?

A)	SI	<input type="checkbox"/>
B)	NO	

¿Por qué?

PREGUNTA 3

Según la metodología MCM propuesta. ¿Considera usted correcto que la elaboración del modelo metodológico describe mediante las actividades a realizar, un nivel de abstracción y relación comprensible para todo el proceso de la metodología MCM propuesta?

A)	SI	<input type="checkbox"/>
B)	NO	<input type="checkbox"/>

¿Por qué?

PREGUNTA 4

Según la metodología MCM propuesta. ¿Considera usted correcto que la elaboración de un análisis de factibilidad de migración de una arquitectura monolítica a una basada en microservicios basado en un cuestionario puede establecer un juicio correcto para validar la "factibilidad de migración" en la metodología MCM propuesta?

A)	SI	<input type="checkbox"/>
B)	NO	<input type="checkbox"/>

¿Por qué?

PREGUNTA 5

Según la metodología MCM propuesta. ¿Considera usted correcto que la actividad de planeamiento de software se haga uso de una metodología ágil como Scrum, XP, con el fin de hacer mejoras continuas en el desarrollo de software?

A)	SI	<input type="checkbox"/>
B)	NO	<input type="checkbox"/>

¿Por qué?

PREGUNTA 6

Según la metodología MCM propuesta. ¿Considera usted que para la actividad de diseño de software es correcto hacer un análisis de la tecnología actual y la tecnología a migrar a través de un cuadro comparativo para tener un concepto general de las ventajas y desventajas de las mismas?

A)	SI	<input type="checkbox"/>
B)	NO	

¿Por qué?

PREGUNTA 7

Según la metodología MCM propuesta. ¿Considera usted correcto hacer uso de un modelado con base en BPMN, con la finalidad de tener un mejor concepto del proceso a realizar?

A)	SI	<input type="checkbox"/>
B)	NO	

¿Por qué?

PREGUNTA 8

Según la metodología MCM propuesta. ¿Considera usted correcto que hacer uso del enfoque Domain Driven Design (DDD) ayuda a los involucrados en el proceso de migración a manejar un mejor lenguaje ubicuo?

A)	SI	<input type="checkbox"/>
B)	NO	

¿Por qué?

PREGUNTA 9

Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI retiraría/adicionaría algún elemento del modelo metodológico, para generar una metodología aplicada a una pequeña empresa?

A)	SI	
B)	NO	

¿Por qué?

--

PREGUNTA 10

Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI cree que la metodología MCM a través de sus 5 actividades disminuye la complejidad de migración de arquitectura a microservicios?

A)	SI	
B)	NO	

¿Por qué?

--

Aquí culminan sus respuestas a este cuestionario.

Siguiendo el método Delphi, se le volverá a enviar este documento con sus respuestas, agregando las respuestas de los otros evaluadores para que sirvan de realimentación o reflexión a sus primeras respuestas y así sus respuestas sean mejoradas o reafirmadas. Esto permitirá una afinación y consenso de respuestas por parte de los evaluadores.

Muchas gracias por su ayuda

Jersson German Mendoza Linares

Anexo 3.4. Comunicación con los expertos

Primera iteración

Experto 01

Apoyo para la evaluación de la metodología MCM - primera opinión

JERSSÓN GERMÁN MENDOZA LINARES <mlinaresjersson@crece.uss.edu.pe>
para dejofua

lun, 8 nov 20:40 (hace 9 días)

Buenas tardes Ing. Denny Fuentes Adrianzén

Reciba un cordial saludo, le escribe Jersson Germán Mendoza Linares, estudiante del X ciclo de la carrera de Ingeniería de sistemas de la Universidad Señor de Sipán - Chiclayo -Perú.

Mediante el presente me dirijo a usted para solicitar su apoyo en una evaluación de mi Metodología de Conversión de Aplicaciones Monolíticas a Microservicios Desplegadas en la Nube para Pequeñas Empresas, que propongo en mi tesis.

La evaluación consta en responder un cuestionario de 10 preguntas. La metodología aplicada en esta evaluación es el método Delphi, por lo que sus respuestas serán mantenidas en el anonimato, esto servirá para mejorar las respuesta en el segundo llenado del mismo cuestionario para los otros dos evaluadores.

El procedimiento del segundo llenado también aplicará a usted y recibirá las respuestas de los demás evaluadores.

Espero su apoyo, adjunto el Resumen de mi Metodología de Conversión de Aplicaciones Monolíticas a Microservicios Desplegadas en la Nube para Pequeñas Empresa y el cuestionario. Tanto si desea no participar, rogaría me notifique su decisión.

Muchas gracias por su tiempo.

Atte. Jersson German Mendoza Linares

2 archivos adjuntos

- cuestionario para e...
- resumen para el m...

Denny John Fuentes Adrianzén
para mí

lun, 6 dic, 23:10 (hace 23 horas)

Saludos Cordiales.
Adjunto lo solicitado.
Atentamente.
Ing. Denny Fuentes A.

De: JERSSÓN GERMÁN MENDOZA LINARES <mlinaresjersson@crece.uss.edu.pe>
Enviado: martes, 7 de diciembre de 2021 3:42
Para: dejofua@hotmail.com <dejofua@hotmail.com>
Asunto: Apoyo para la evaluación de la metodología MCM - primera opinión


[Mensaje recortado] [Ver todo el mensaje](#)

2 archivos adjuntos

- 01 Resumen para e...
- 01 Cuestionario pa...

Experto 02

Apoyo para la evaluación de la metodología MCM - primera opinión Recibidos x

 **JERSSÓN GERMÁN MENDOZA LINARES** <mlinaresjersson@crece.uss.edu.pe>
para cmacojunio

lun, 8 nov 20:42 (hace 9 días) ★ ↶ ⋮

Buenas tardes Ing. Junior Eugenio Cachay Maco

Reciba un cordial saludo, le escribe Jerssón Germán Mendoza Linares, estudiante del X ciclo de la carrera de ingeniería de sistemas de la Universidad Señor de Sipán - Chiclayo -Perú.

Mediante el presente me dirijo a usted para solicitar su apoyo en una evaluación de mi Metodología de Conversión de Aplicaciones Monolíticas a Microservicios Desplegadas en la Nube para Pequeñas Empresas, que propongo en mi tesis.

La evaluación consta en responder un cuestionario de 10 preguntas. La metodología aplicada en esta evaluación es el método Delphi, por lo que sus respuestas serán mantendrán en el anonimato, esto servirá para mejorar las respuesta en el segundo llenado del mismo cuestionario para los otros dos evaluadores.



El procedimiento del segundo llenado también aplicará a usted y recibirá las respuestas de los demás evaluadores.

Espero su apoyo, adjunto el Resumen de mi Metodología de Conversión de Aplicaciones Monolíticas a Microservicios Desplegadas en la Nube para Pequeñas Empresa y el cuestionario. Tanto si desea no participar, rogaría me notifique su decisión.

Muchas gracias por su tiempo.

Atte. Jerssón German Mendoza Linares

2 archivos adjuntos

 **JUNIOR EUGENIO CACHAY MACO**
para mí


sáb, 13 nov 10:58 (hace 8 días) ☆ ↶ ⋮

Estimado, buen día.
Adjunto lo solicitados.
Saludos
Jr.



Experto 03

Apoyo para la evaluacion de la metodologia MCM - primera opinion Externo Recibidos x

 **JERSSÓN GERMÁN MENDOZA LINARES** <mlinaresjersson@crece.uss.edu.pe>
para karlosvaldivia

lun, 8 nov 20:43 (hace 9 días) ★ ↶ ⋮

Buenas tardes Ing. Carlos Valdivia Salazar

Reciba un cordial saludo, le escribe Jersson Germán Mendoza Linares, estudiante del X ciclo de la carrera de ingeniería de sistemas de la Universidad Señor de Sipán - Chiclayo -Perú.

Mediante el presente me dirijo a usted para solicitar su apoyo en una evaluación de mi Metodología de Conversión de Aplicaciones Monolíticas a Microservicios Desplegadas en la Nube para Pequeñas Empresas, que propongo en mi tesis.

La evaluación consta en responder un cuestionario de 10 preguntas. La metodología aplicada en esta evaluación es el método Delphi, por lo que sus respuestas serán mantendrán en el anonimato, esto servirá para mejorar las respuesta en el segundo llenado del mismo cuestionario para los otros dos evaluadores.



El procedimiento del segundo llenado también aplicará a usted y recibirá las respuestas de los demás evaluadores.

Espero su apoyo, adjunto el Resumen de mi Metodología de Conversión de Aplicaciones Monolíticas a Microservicios Desplegadas en la Nube para Pequeñas Empresa y el cuestionario. Tanto si desea no participar, rogaría me notifique su decisión.

Muchas gracias por su tiempo.

Atte. Jersson German Mendoza Linares

2 archivos adjuntos

 cuestionario para e...  resumen para el m...


 **Ing. Carlos Valdivia Salazar**
para mi


mar, 16 nov 1:00 (hace 5 días) ☆ ↶ ⋮

Estimado Jersson:

Envío las respuestas al cuestionario de evaluación.

Saludos cordiales.

 Libre de virus. www.avast.com

 cuestionario para e...

Segunda iteración

Experto 01

Apoyo para la evaluación de la metodología MCM - segunda opinión

JERSSÓN GERMÁN MENDOZA LINARES <mlianaresjersson@crece.uss.edu.pe>
para dejofua

0:47 (hace 16 minutos) ☆ ↶ ⋮

Buenas noches Ing. Denny Fuentes Adriánzén

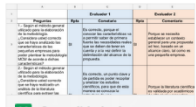
Reciba un cordial saludo y agradecimiento por el apoyo brindado para colaborar con mi investigación

Mediante el presente me dirijo a usted para solicitar nuevamente su apoyo en la segunda iteración de opiniones, se adjunta una tabla donde se consolidó las respuestas de los evaluadores, usted se encuentra identificado como "Evaluador 1" y sus respuestas estan en una columna de color azul.

Si sus respuestas tienden a tener un cambio al leer las otras respuestas de los evaluadores, lo debe de indicar el cambio de respuesta y el por qué en la columna que se llama "Segunda opinion", la cual se encuentra de color amarillo. Asimismo, si mantiene su respuesta deberá de colocar "Conservo mi opinión"

Muchas gracias por su tiempo.

Atte. Jersson German Mendoza Linares



Consolidado de res...

Denny John Fuentes Adriánzén <dejofua@hotmail.com>
para mí

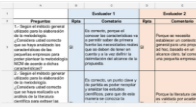
mié, 12 ene, 10:14 ☆ ↶

Envío lo solicitado.
Atte.
Ing. Denny Fuentes A.

De: JERSSÓN GERMÁN MENDOZA LINARES <mlianaresjersson@crece.uss.edu.pe>
Enviado: viernes, 10 de diciembre de 2021 5:47
Para: dejofua@hotmail.com <dejofua@hotmail.com>
Asunto: Apoyo para la evaluación de la metodología MCM - segunda opinión



...





[Mensaje recortado] [Ver todo el mensaje](#)



Consolidado de res...

Experto 02

Apoyo para la evaluación de la metodología MCM - segunda opinión  

 **JERSSÓN GERMÁN MENDOZA LINARES** <mлинаresjersson@crece.uss.edu.pe>
para JUNIOR ▾ 0:51 (hace 13 minutos)   

Buenas noches Ing. Junior Eugenio Cachay Maco

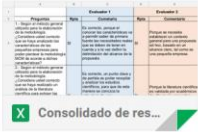
Reciba un cordial saludo y agradecimiento por el apoyo brindado para colaborar con mi investigación


Mediante el presente me dirijo a usted para solicitar nuevamente su apoyo en la segunda iteración de opiniones, se adjunta una tabla donde se consolidó las respuestas de los evaluadores, usted se encuentra identificado como "Evaluador 2" y sus respuestas están en una columna de color naranja.

Si sus respuestas tienden a tener un cambio al leer las otras respuestas de los evaluadores, lo debe de indicar el cambio de respuesta y el por qué en la columna que se llama "Segunda opinión", la cual se encuentra de color amarillo. Asimismo, si mantiene su respuesta deberá de colocar "Conservo mi opinión"

Muchas gracias por su tiempo.

Atte. Jersson German Mendoza Linares



 Consolidado de res...

 **JUNIOR EUGENIO CACHAY MACO** <cmacojunio@crece.uss.edu.pe>
para mí ▾ mié, 29 dic 2021, 23:43  

Estimado Jersson, adjunto lo solicitado.

Saludos

Jr






 Consolidado de res...

Experto 03

Apoyo para la evaluación de la metodología MCM - segunda opinión

 **JERSSÓN GERMÁN MENDOZA LINARES** <mlinaresjersson@crece.uss.edu.pe>
para karlosvaldivia

0:49 (hace 16 minutos) ☆ ↶ ⋮

Buenas noches Ing. Carlos Valdivia Salazar

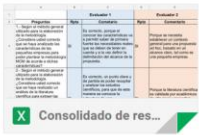
Reciba un cordial saludo y agradecimiento por el apoyo brindado para colaborar con mi investigación

Mediante el presente me dirijo a usted para solicitar nuevamente su apoyo en la segunda iteración de opiniones, se adjunta una tabla donde se consolidó las respuestas de los evaluadores, usted se encuentra identificado como "Evaluador 3" y sus respuestas están en una columna de color verde.


Si sus respuestas tienden a tener un cambio al leer las otras respuestas de los evaluadores, lo debe de indicar el cambio de respuesta y el por qué en la columna que se llama "Segunda opinión", la cual se encuentra de color amarillo. Asimismo, si mantiene su respuesta deberá de colocar "Conservo mi opinión"

Muchas gracias por su tiempo.

Atte. Jerssón German Mendoza Linares



Consolidado de res...

 **Ing. Carlos Valdivia Salazar** <karlosvaldivia@gmail.com>
para mí

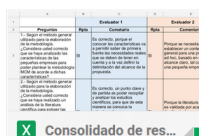
mié, 15 dic 2021, 9:59 ☆ ↶

Buenos días

Envío las respuestas al instrumento enviado.

Saludos.

...



Consolidado de res...

Anexo 3.5. Cuestionario para los integrantes de la ejecución del caso práctico de la MCM en el aplicativo “GMO” en HTEC E.I.R.L.

CUESTIONARIO PARA LA MEDICIÓN DEL NIVEL DE SATISFACCION, PRODUCTIVIDAD Y COMPLEJIDAD CON BASE EN LA PROPUESTA DE METODOLOGÍA DE CONVERSIÓN DE APLICACIONES MONOLÍTICAS A MICROSERVICIOS DESPLEGABLE EN LA NUBE PARA PEQUEÑAS EMPRESAS ELABORADA POR JERSSON GERMAN MENDOZA LINARES

Consideraciones:

- Las respuestas se mantendrán en el anonimato bajo el conocimiento del investigador.
- Complete con una "X" en alguna de las alternativas o llene el recuadro según se le indique, luego se solicita redactar su respuesta a la pregunta "¿Por qué?" en las casillas de color celeste

PREGUNTA 1

Según la metodología MCM propuesta. ¿En qué porcentaje cree usted que su nivel de productividad haya mejorado en el proceso de desarrollo e implementación de nuevos requerimientos características de la aplicación orientada a microservicios GMO++, a diferencia de la aplicación monolítica GMO?

1	Porcentaje de productividad de 1 – 100%	
---	---	--

¿Por qué?

PREGUNTA 2

Según la metodología MCM propuesta, del 1 al 5 ¿Cuál es el nivel de satisfacción en cuanto al proceso de implementación de la metodología MCM en la aplicación GMO?

1	Muy insatisfecho	
2	Insatisfecho	
3	Regular	
4	Satisfecho	
5	Muy satisfecho	

¿Por qué?

PREGUNTA 3

Según la metodología MCM propuesta, del 1 al 5 ¿Cuál es el nivel de complejidad en cuanto al proceso de implementación de la metodología MCM en la aplicación GMO?

1	Muy baja	
2	Baja	
3	Regular	
4	Alta	
5	Muy alta	

¿Por qué?

Aquí culminan sus respuestas a este cuestionario.

Muchas gracias por su ayuda

Jersson German Mendoza Linares

Encuestado 01

**CUESTIONARIO PARA LA MEDICIÓN EL NIVEL DE SATISFACCION Y PRODUCTIVIDAD CON BASE EN
PROPUESTA DE METODOLOGÍA DE CONVERSIÓN DE APLICACIONES MONOLÍTICAS A
MICROSERVICIOS DESPLEGABLE EN LA NUBE PARA PEQUEÑAS EMPRESAS ELABORADA POR
JERSSON GERMAN MENDOZA LINARES**

Consideraciones:

- Las respuestas se mantendrán en el anonimato bajo el conocimiento del investigador.
- Complete con una "X" en alguna de las alternativas o llene el recuadro según se le indique, luego se solicita redactar su respuesta a la pregunta "¿Por qué?" en las casillas de color celeste

PREGUNTA 1

Según la metodología MCM propuesta. ¿En qué porcentaje cree usted que su nivel de productividad haya mejorado en el proceso de desarrollo e implementación de nuevos requerimientos características de la aplicación orientada a microservicios GMO++, a diferencia de la aplicación monolítica GMO?

1	Porcentaje de productividad de 1 – 100%	23%
---	---	------------

¿Por qué?

Por qué solo me concentraba en un solo proceso a la vez, mientras los demás desarrolladores avanzaban otras características.

PREGUNTA 2

Según la metodología MCM propuesta, del 1 al 5 ¿Cuál es el nivel de satisfacción en cuanto al proceso de implementación de la metodología MCM en la aplicación GMO?

1	Muy insatisfecho	X
2	Insatisfecho	
3	Regular	
4	Satisfecho	
5	Muy satisfecho	

¿Por qué?

Por qué fue muy interesante el aprender e implementar una metodología diferente.

PREGUNTA 3

Según la metodología MCM propuesta, del 1 al 5 ¿Cuál es el nivel de complejidad en cuanto al proceso de implementación de la metodología MCM en la aplicación GMO?

1	Muy baja	
2	Baja	X
3	Regular	
4	Alta	
5	Muy alta	

¿Por qué?

Por qué se entendió el proceso que se debía realizar.

Aquí culminan sus respuestas a este cuestionario.

Muchas gracias por su ayuda

Jersson German Mendoza Linares

Encuestado 02

**CUESTIONARIO PARA LA MEDICIÓN EL NIVEL DE SATISFACCION Y PRODUCTIVIDAD CON BASE EN
PROPUESTA DE METODOLOGÍA DE CONVERSIÓN DE APLICACIONES MONOLÍTICAS A
MICROSERVICIOS DESPLEGADAS EN LA NUBE PARA PEQUEÑAS EMPRESAS ELABORADA POR
JERSSON GERMAN MENDOZA LINARES**

Consideraciones:

- Las respuestas se mantendrán en el anonimato bajo el conocimiento del investigador.
- Complete con una "X" en alguna de las alternativas o llene el recuadro según se le indique, luego se solicita redactar su respuesta a la pregunta "¿Por qué?" en las casillas de color celeste

PREGUNTA 1

Según la metodología MCM propuesta. ¿En qué porcentaje cree usted que su nivel de productividad haya mejorado en el proceso de desarrollo e implementación de nuevos requerimientos características de la aplicación orientada a microservicios GMO++, a diferencia de la aplicación monolítica GMO?

1	Porcentaje de productividad de 1 – 100%	15%
---	---	-----

¿Por qué?

Por qué desarrollaba una característica, cuando antes en la misma base de código realizaba todo el proceso.

PREGUNTA 2

Según la metodología MCM propuesta, del 1 al 5 ¿Cuál es el nivel de satisfacción en cuanto al proceso de implementación de la metodología MCM en la aplicación GMO?

1	Muy insatisfecho	X
2	Insatisfecho	
3	Regular	
4	Satisfecho	
5	Muy satisfecho	

¿Por qué?

Por qué el proceso de implementación se realizo correctamente.

PREGUNTA 3

Según la metodología MCM propuesta, del 1 al 5 ¿Cuál es el nivel de complejidad en cuanto al proceso de implementación de la metodología MCM en la aplicación GMO?

1	Muy baja	X
2	Baja	
3	Regular	
4	Alta	
5	Muy alta	

¿Por qué?

Por que se entendi6 la metodologí.

Aquí culminan sus respuestas a este cuestionario.

Muchas gracias por su ayuda

Jersson German Mendoza Linares

Encuestado 03

**CUESTIONARIO PARA LA MEDICIÓN EL NIVEL DE SATISFACCION Y PRODUCTIVIDAD CON BASE EN
PROPUESTA DE METODOLOGÍA DE CONVERSIÓN DE APLICACIONES MONOLÍTICAS A
MICROSERVICIOS DESPLEGADAS EN LA NUBE PARA PEQUEÑAS EMPRESAS ELABORADA POR
JERSSON GERMAN MENDOZA LINARES**

Consideraciones:

- Las respuestas se mantendrán en el anonimato bajo el conocimiento del investigador.
- Complete con una "X" en alguna de las alternativas o llene el recuadro según se le indique, luego se solicita redactar su respuesta a la pregunta "¿Por qué?" en las casillas de color celeste

PREGUNTA 1

Según la metodología MCM propuesta. ¿En qué porcentaje cree usted que su nivel de productividad haya mejorado en el proceso de desarrollo e implementación de nuevos requerimientos características de la aplicación orientada a microservicios GMO++, a diferencia de la aplicación monolítica GMO?

1	Porcentaje de productividad de 1 – 100%	20%
---	---	------------

¿Por qué?

Por qué los microservicios eras desarrollados independientemente y en paralelo.

PREGUNTA 2

Según la metodología MCM propuesta, del 1 al 5 ¿Cuál es el nivel de satisfacción en cuanto al proceso de implementación de la metodología MCM en la aplicación GMO?

1	Muy insatisfecho	X
2	Insatisfecho	
3	Regular	
4	Satisfecho	
5	Muy satisfecho	

¿Por qué?

Por qué se aprendieron nuevas cosas durante la implementación.

PREGUNTA 3

Según la metodología MCM propuesta, del 1 al 5 ¿Cuál es el nivel de complejidad en cuanto al proceso de implementación de la metodología MCM en la aplicación GMO?

1	Muy baja	
2	Baja	
3	Regular	X
4	Alta	
5	Muy alta	

¿Por qué?

Por qué al principio fue complejo, pero después se comprendió el proceso metodológico.

Aquí culminan sus respuestas a este cuestionario.

Muchas gracias por su ayuda

Jersson German Mendoza Linares

Encuestado 04

**CUESTIONARIO PARA LA MEDICIÓN EL NIVEL DE SATISFACCION Y PRODUCTIVIDAD CON BASE EN
PROPUESTA DE METODOLOGÍA DE CONVERSIÓN DE APLICACIONES MONOLÍTICAS A
MICROSERVICIOS DESPLEGADAS EN LA NUBE PARA PEQUEÑAS EMPRESAS ELABORADA POR
JERSSON GERMAN MENDOZA LINARES**

Consideraciones:

- Las respuestas se mantendrán en el anonimato bajo el conocimiento del investigador.
- Complete con una "X" en alguna de las alternativas o llene el recuadro según se le indique, luego se solicita redactar su respuesta a la pregunta "¿Por qué?" en las casillas de color celeste

PREGUNTA 1

Según la metodología MCM propuesta. ¿En qué porcentaje cree usted que su nivel de productividad haya mejorado en el proceso de desarrollo e implementación de nuevos requerimientos características de la aplicación orientada a microservicios GMO++, a diferencia de la aplicación monolítica GMO?

1	Porcentaje de productividad de 1 – 100%	25%
---	---	-----

¿Por qué?

Por qué a diferencia de la aplicación monolítica, se desarrollaron de manera independiente los microservicios dividiendo las tareas.

PREGUNTA 2

Según la metodología MCM propuesta, del 1 al 5 ¿Cuál es el nivel de satisfacción en cuanto al proceso de implementación de la metodología MCM en la aplicación GMO?

1	Muy insatisfecho	X
2	Insatisfecho	
3	Regular	
4	Satisfecho	
5	Muy satisfecho	

¿Por qué?

Por qué la implementación se desarrollo de manera correcta.

PREGUNTA 3

Según la metodología MCM propuesta, del 1 al 5 ¿Cuál es el nivel de complejidad en cuanto al proceso de implementación de la metodología MCM en la aplicación GMO?

1	Muy baja	X
2	Baja	
3	Regular	
4	Alta	
5	Muy alta	

¿Por qué?

Por qué se comprendió la metodología.

Aquí culminan sus respuestas a este cuestionario.

Muchas gracias por su ayuda

Jersson German Mendoza Linares

Encuestado 05

**CUESTIONARIO PARA LA MEDICIÓN EL NIVEL DE SATISFACCIÓN Y PRODUCTIVIDAD CON BASE EN
PROPUESTA DE METODOLOGÍA DE CONVERSIÓN DE APLICACIONES MONOLÍTICAS A
MICROSERVICIOS DESPLEGADAS EN LA NUBE PARA PEQUEÑAS EMPRESAS ELABORADA POR
JERSSON GERMAN MENDOZA LINARES**

Consideraciones:

- Las respuestas se mantendrán en el anonimato bajo el conocimiento del investigador.
- Complete con una "X" en alguna de las alternativas o llene el recuadro según se le indique, luego se solicita redactar su respuesta a la pregunta "¿Por qué?" en las casillas de color celeste

PREGUNTA 1

Según la metodología MCM propuesta. ¿En qué porcentaje cree usted que su nivel de productividad haya mejorado en el proceso de desarrollo e implementación de nuevos requerimientos características de la aplicación orientada a microservicios GMO++, a diferencia de la aplicación monolítica GMO?

1	Porcentaje de productividad de 1 – 100%	18%
---	---	-----

¿Por qué?

Por qué durante el desarrollo se avanzaron las características con facilidad, al contrario que con el aplicativo monolítico.

PREGUNTA 2

Según la metodología MCM propuesta, del 1 al 5 ¿Cuál es el nivel de satisfacción en cuanto al proceso de implementación de la metodología MCM en la aplicación GMO?

1	Muy insatisfecho	X
2	Insatisfecho	
3	Regular	
4	Satisfecho	
5	Muy satisfecho	

¿Por qué?

Por qué en el proceso de desarrollo se aprendió sobre la arquitectura.

PREGUNTA 3

Según la metodología MCM propuesta, del 1 al 5 ¿Cuál es el nivel de complejidad en cuanto al proceso de implementación de la metodología MCM en la aplicación GMO?

1	Muy baja	X
2	Baja	
3	Regular	
4	Alta	
5	Muy alta	

¿Por qué?

Por qué el proceso de implementación se comprendió de la mejor manera..

Aquí culminan sus respuestas a este cuestionario.

Muchas gracias por su ayuda

Jersson German Mendoza Linares0

Anexo 4. Respuestas y comentarios del cuestionario de los expertos del metodo Delphi.

Primera iteración del metodo Delphi.

Preguntas	Evaluador 1		Evaluador 2		Evaluador 3	
	Rpta	Comentario	Rpta	Comentario	Rpta	Comentario
<p>1.- Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya analizado las características de las pequeñas empresas para poder plantear la metodología MCM de acorde a dichas características?</p>	SI	Es correcto, porque el conocer las características va a permitir saber de primera fuente las necesidades reales que se deben de tener en cuenta y a la vez definir la delimitación del alcance de la propuesta.	SI	Porque se necesita establecer un contexto general para una propuesta ad hoc, basado en un alcance claro, tal como es una pequeña empresa.	SI	Es necesario identificar las tecnologías con las que cuentan los aplicativos monolíticos.
<p>2.- Según el método generar utilizado para la elaboración de la metodología. ¿Considera usted</p>	SI	Es correcto, un punto clave y de partida es poder recopilar y analizar los estudios científicos, para que de esta manera se conozca la estructura	SI	Porque la literatura científica es validada por académicos de alto nivel y presentan cuidado en la	SI	Es necesario conocer el ámbito actual de las metodologías que tratan sobre el tema de microservicios, y poder

<p>correcto que se haya realizado un análisis de la literatura científica para extraer las características, herramientas más utilizadas en la arquitectura basada en microservicios y así poder tener un concepto general para el desarrollo de la MCM propuesta?</p>		<p>de una arquitectura orientada a microservicios y en función a ello plantear una alternativa acorde a las necesidades identificadas.</p>		<p>calidad de producción científica.</p>	<p>realizar una nueva propuesta.</p>
<p>3.- Según la metodología MCM propuesta. ¿Considera usted correcto que la elaboración del modelo metodológico describe mediante las actividades a realizar, un nivel de abstracción y</p>	<p>SI</p>	<p>Es correcto, en el modelo planteado se define todo el conjunto de actividades y tareas que deben de ejecutarse que permitan identificar las características propias que se sigue en el proceso de la metodología en todo su ciclo de vida de desarrollo.</p>	<p>SI</p>	<p>Porque estás considerando verificación por expertos y validación con un caso de estudio.</p>	<p>SI Abarca todas las etapas necesarias para descomponer y desacoplar la aplicación monolítica en una serie de microservicios.</p>

<p>relación comprensible para todo el proceso de la metodología MCM propuesta?</p>						
<p>4.- Según la metodología MCM propuesta. ¿Considera usted correcto que la elaboración de un análisis de factibilidad de migración de una arquitectura monolítica a una basada en microservicios basado en un cuestionario puede establecer un juicio correcto para validar la “factibilidad de migración” en la metodología MCM propuesta?</p>	<p>SI</p>	<p>Es Correcto, el análisis de la factibilidad va a permitir conocer si es posible hacer el proceso de migración conociendo los aspectos técnicos, económicos y operativos a considerarse que garanticen la fiabilidad del modelo a desarrollar.</p>	<p>NO</p>	<p>Porque estás considerando que el análisis lo realice un Gerente con sólo preguntas. Esto amerita que:</p> <ol style="list-style-type: none"> 1. El Gerente tenga un alto nivel de experiencia en operatividad. 2. Se obvien algunos aspectos técnicos. 	<p>SI</p>	<p>Es necesario conocer si es factible convertir una aplicación monolítica a microservicios.</p>

<p>5.- Según la metodología MCM propuesta. ¿Considera usted correcto que la actividad de planeamiento de software se haga uso de una metodología ágil como Scrum, XP, con el fin de hacer mejoras continuas en el desarrollo de software?</p>	<p>SI</p>	<p>Es correcto, el objetivo de un paradigma ágil es poder reducir los tiempos de desarrollo y tener en contacto directo a los usuarios en cada iteración que se tiene que realizar, teniendo en cuenta que un punto de partida es poder considerar las necesidades o requerimientos de usuarios.</p>	<p>SI</p>	<p>Porque para pequeñas empresas se necesita agilidad.</p>	<p>SI</p> <p>Como la metodología está basada en las fases de la ingeniería del software, es necesario el uso de una metodología ágil para que acelere el proceso.</p>
<p>6.- Según la metodología MCM propuesta. ¿Considera usted que para la actividad de diseño de software es correcto hacer un análisis de la tecnología actual y la tecnología a migrar a través de un cuadro</p>	<p>SI</p>	<p>Es correcto, la actividad de Diseño es uno de los puntos críticos, es decir, en ella se plantea el modelo de propuesta considerando la evaluación del paradigma actual y el propuesto, para ello debe de organizarse un comparativos de factores de evaluación.</p>	<p>SI</p>	<p>Porque orientará al proceso de reingeniería y en el uso de recursos sólidos ya existentes.</p>	<p>SI</p> <p>Es necesario realizar este cuadro comparativo, sin embargo, es necesario que se cuente con expertos en el área para identificar estas ventajas y desventajas.</p>

comparativo para tener un concepto general de las ventajas y desventajas de las mismas?						
7.- Según la metodología MCM propuesta. ¿Considera usted correcto hacer uso de un modelado con base en BPMN, con la finalidad de tener un mejor concepto del proceso a realizar?	SI	Es correcto, considerar que el modelar los procesos de negocios nos va poder conocer la integridad y relacionamiento de cada uno de los elementos interactuantes del sistema, a la vez que se puede generar una simulación previa del modelo y poder evaluar sus efectos y sensibilidad como parte del sistema.	SI	Porque representa de manera práctica el flujo de información que sostendrá la solución tecnológica propuesta.	SI	El manejo de base de datos es importante en todo desarrollo de software, y en el caso de microservicios mucho más.
8.- Según la metodología MCM propuesta. ¿Considera usted correcto que hacer uso del enfoque Domain Driven	SI	Es correcto, El Diseño del Dominio, va a permitir definir claramente cuál es el alcance que debe de tenerse, además que con ello se puede	SI	Porque es esencial manejar un lenguaje uniforme de comunicación para reforzar el conocimiento de desarrollo, y evitar así problemas ante	SI	Teniendo que DDD considera al conocimiento del dominio para crear una aplicación, es importante contar con un lenguaje entre las personas que

Design (DDD) ayuda a los involucrados en el proceso de migración a manejar un mejor lenguaje ubicuo?		delimitar correctamente los costos, tiempos y riesgos del desarrollo en base al modelo propuesto.		rotación o renuncia de personal del proyecto.		conocen el proceso y los expertos.
9.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI retiraría/adicionaría algún elemento del modelo metodológico, para generar una metodología aplicada a una pequeña empresa?	NO	Considerar que se está proponiendo un modelo y la idea es poder evaluar sus resultados y en función a ello, poder hacer los ajustes necesarios que garanticen un mejor soporte a los objetivos propuestos como alternativa de solución.	SI	Porque aún falta mejorar actividades de verificación y validación de software (V&V), si bien no se necesita una gama amplia de pruebas V&V, al menos lo básico para una pequeña empresa.	NO	Están consideradas las etapas necesarias para un modelo de migración de una aplicación monolítica a microservicios.
10.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI cree que la metodología MCM a través de sus 5 actividades	SI	Es correcto, es importante es poder conocer paso a paso cada una de las actividades y más aún si el proceso de abstracción es por módulos o etapas de	SI	Porque abarca muchos aspectos importantes para la migración ordenada.	SI	Está basada en un análisis de las tecnologías de la aplicación monolítica y la propuesta del uso de nuevas tecnologías hacia los microservicios.

disminuye la complejidad de migración de arquitectura a microservicios?		esta manera se puede identificar cada uno de los detalles de los elementos intervinientes.				
--	--	--	--	--	--	--

Segunda iteración del metodo Delphi.

Preguntas	Evaluador 1		Evaluador 2		Evaluador 3	
	Rpta	Comentario	Rpta	Comentario	Rpta	Comentario
<p>1.- Según el método general utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya analizado las características de las pequeñas empresas para poder plantear la metodología MCM de acorde a dichas características?</p>	SI	Es correcto, porque el conocer las características va a permitir saber de primera fuente las necesidades reales que se deben de tener en cuenta y a la vez definir la delimitación del alcance de la propuesta. (Conservó su opinión)	SI	Porque se necesita establecer un contexto general para una propuesta ad hoc, basado en un alcance claro, tal como es una pequeña empresa. (Conservó su opinión)	SI	Es necesario identificar las tecnologías con las que cuentan los aplicativos monolíticos. (Conservó su opinión)
<p>2.- Según el método generar utilizado para la elaboración de la metodología. ¿Considera usted correcto que se haya realizado un</p>	SI	Es correcto, un punto clave y de partida es poder recopilar y analizar los estudios científicos, para que de esta manera se conozca la estructura de una arquitectura orientada a	SI	Porque la literatura científica es validada por académicos de alto nivel y presentan cuidado en la calidad de producción científica. (Conservó su opinión)	SI	Es necesario conocer el ámbito actual de las metodologías que tratan sobre el tema de microservicios, y poder realizar una nueva

<p>análisis de la literatura científica para extraer las características, herramientas más utilizadas en la arquitectura basada en microservicios y así poder tener un concepto general para el desarrollo de la MCM propuesta?</p>		<p>microservicios y en función a ello plantear una alternativa acorde a las necesidades identificadas. (Conservó su opinión)</p>			<p>propuesta. (Conservó su opinión)</p>
<p>3.- Según la metodología MCM propuesta. ¿Considera usted correcto que la elaboración del modelo metodológico describe mediante las actividades a realizar, un nivel de abstracción y relación comprensible para</p>	<p>SI</p>	<p>Es correcto, en el modelo planteado se define todo el conjunto de actividades y tareas que deben de ejecutarse que permitan identificar las características propias que se sigue en el proceso de la metodología en todo su ciclo de vida de desarrollo. (Conservó su opinión)</p>	<p>SI</p>	<p>Porque estás considerando verificación por expertos y validación con un caso de estudio. (Conservó su opinión)</p>	<p>SI Abarca todas las etapas necesarias para descomponer y desacoplar la aplicación monolítica en una serie de microservicios. (Conservó su opinión)</p>

todo el proceso de la metodología MCM propuesta?						
4.- Según la metodología MCM propuesta. ¿Considera usted correcto que la elaboración de un análisis de factibilidad de migración de una arquitectura monolítica a una basada en microservicios basado en un cuestionario puede establecer un juicio correcto para validar la “factibilidad de migración” en la metodología MCM propuesta?	SI	Es Correcto, el análisis de la factibilidad va a permitir conocer si es posible hacer el proceso de migración conociendo los aspectos técnicos, económicos y operativos a considerarse que garanticen la fiabilidad del modelo a desarrollar. (Conservó su opinión)	NO	Porque estás considerando que el análisis lo realice un Gerente con sólo preguntas. Esto amerita que: 1. El Gerente tenga un alto nivel de experiencia en operatividad. 2. Se obvien algunos aspectos técnicos. (Conservó su opinión)	SI	Es necesario conocer si es factible convertir una aplicación monolítica a microservicios. (Conservó su opinión)
5.- Según la metodología MCM propuesta.	SI	Es correcto, el objetivo de un paradigma ágil es poder reducir los tiempos	SI	Porque para pequeñas empresas se necesita	SI	Como la metodología está basada en las fases de la ingeniería del

<p>¿Considera usted correcto que la actividad de planeamiento de software se haga uso de una metodología ágil como Scrum, XP, con el fin de hacer mejoras continuas en el desarrollo de software?</p>		<p>de desarrollo y tener en contacto directo a los usuarios en cada iteración que se tiene que realizar, teniendo en cuenta que un punto de partida es poder considerar las necesidades o requerimientos de usuarios. (Conservó su opinión)</p>		<p>agilidad. (Conservó su opinión)</p>	<p>software, es necesario el uso de una metodología ágil para que acelere el proceso. (Conservó su opinión)</p>
<p>6.- Según la metodología MCM propuesta. ¿Considera usted que para la actividad de diseño de software es correcto hacer un análisis de la tecnología actual y la tecnología a migrar a través de un cuadro comparativo para tener un concepto general de las</p>	<p>SI</p>	<p>Es correcto, la actividad de Diseño es uno de los puntos críticos, es decir, en ella se plantea el modelo de propuesta considerando la evaluación del paradigma actual y el propuesto, para ello debe de organizarse un comparativos de factores de evaluación. (Conservó su opinión)</p>	<p>SI</p>	<p>Porque orientará al proceso de reingeniería y en el uso de recursos sólidos ya existentes. (Conservó su opinión)</p>	<p>SI Es necesario realizar este cuadro comparativo, sin embargo, es necesario que se cuente con expertos en el área para identificar estas ventajas y desventajas. (Conservó su opinión)</p>

ventajas y desventajas de las mismas?						
7.- Según la metodología MCM propuesta. ¿Considera usted correcto hacer uso de un modelado con base en BPMN, con la finalidad de tener un mejor concepto del proceso a realizar?	SI	Es correcto, considerar que el modelar los procesos de negocios nos va poder conocer la integridad y relacionamiento de cada uno de los elementos interactuantes del sistema, a la vez que se puede generar una simulación previa del modelo y poder evaluar sus efectos y sensibilidad como parte del sistema. (Conservó su opinión)	SI	Porque representa de manera práctica el flujo de información que sostendrá la solución tecnológica propuesta. (Conservó su opinión)	SI	El uso de un modelamiento de procesos permite conocer las fases, etapas o actividades y el flujo que se tiene, es muy importante. (Cambió su opinión)
8.- Según la metodología MCM propuesta. ¿Considera usted correcto que hacer uso del enfoque Domain Driven Design (DDD) ayuda a los	SI	Es correcto, El Diseño del Dominio, va a permitir definir claramente cuál es el alcance que debe de tenerse, además que con ello se puede delimitar correctamente los costos, tiempos y	SI	Porque es esencial manejar un lenguaje uniforme de comunicación para reforzar el conocimiento de desarrollo, y evitar así problemas ante rotación o renuncia de personal del	SI	Teniendo que DDD considera al conocimiento del dominio para crear una aplicación, es importante contar con un lenguaje entre las personas que conocen el proceso y los

involucrados en el proceso de migración a manejar un mejor lenguaje ubicuo?		riesgos del desarrollo en base al modelo propuesto. (Conservó su opinión)		proyecto. (Conservó su opinión)		expertos. (Conservó su opinión)
9.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI retiraría/adicionaría algún elemento del modelo metodológico, para generar una metodología aplicada a una pequeña empresa?	NO	Considerar que se está proponiendo un modelo y la idea es poder evaluar sus resultados y en función a ello, poder hacer los ajustes necesarios que garanticen un mejor soporte a los objetivos propuestos como alternativa de solución. (Conservó su opinión)	SI	Porque aún falta mejorar actividades de verificación y validación de software (V&V), si bien no se necesita una gama amplia de pruebas V&V, al menos lo básico para una pequeña empresa. (Conservó su opinión)	NO	Están consideradas las etapas necesarias para un modelo de migración de una aplicación monolítica a microservicios. (Conservó su opinión)
10.- Según la metodología MCM propuesta. ¿Según su experiencia en el área de TI cree que la metodología MCM a través de sus 5 actividades disminuye la complejidad de	SI	Es correcto, es importante es poder conocer paso a paso cada una de las actividades y más aún si el proceso de abstracción es por módulos o etapas de esta manera se puede identificar cada uno de	SI	Porque abarca muchos aspectos importantes para la migración ordenada. (Conservó su opinión)	SI	Está basada en un análisis de las tecnologías de la aplicación monolítica y la propuesta del uso de nuevas tecnologías hacia los microservicios. (Conservó su opinión)

migración de arquitectura a microservicios?		los detalles de los elementos intervinientes. (Conservó su opinión)				
--	--	--	--	--	--	--