



UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS
COMPUTACIONALES

PROYECTO DE INVESTIGACIÓN

**“ESTUDIO METODOLÓGICO PARA EL DESARROLLO DE MICROSERVICIOS,
DISEÑO DE UN PROTOTIPO”**

AUTOR:

Guevara Naranjo Christian Paúl

TUTOR:

Ing. MSC. Verónica Tapia Cerda

Latacunga - Ecuador

Marzo, 2021



DECLARATORIA DE AUTORÍA

Yo, Guevara Naranjo Christian Paúl, con número de C.I: **172299198-9**, declaro ser autor del presente proyecto de investigación: **“ESTUDIO METODOLÓGICO PARA EL DESARROLLO DE MICROSERVICIOS, DISEÑO DE UN PROTOTIPO”**, siendo la **Ing. MSc. Verónica del Consuelo Tapia Cerda** tutora del presente trabajo; y eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Atentamente

.....
Guevara Naranjo Christian Paúl
CI: 172299198-9



AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN

En calidad de Tutor del Proyecto de Investigación sobre el título:

“ESTUDIO METODOLÓGICO PARA EL DESARROLLO DE MICROSERVICIOS, DISEÑO DE UN PROTOTIPO” de **Guevara Naranjo Christian Paúl**, con numero de C.I: **172299198-9**, de la carrera de Ingeniería en Informática y Sistemas Computacionales, considero que dicho Proyecto de Investigación cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Honorable Consejo Académico de la Facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

Latacunga, marzo, 2021

Tutor de Titulación
Ing. MSc. Verónica Tapia Cerda
CC: 050205369-7



APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la FACULTAD de Ciencias de la Ingeniería y Aplicadas; por cuanto, el postulante: **Guevara Naranjo Christian Paúl**, con el título de Proyecto de titulación: **“ESTUDIO METODOLÓGICO PARA EL DESARROLLO DE MICROSERVICIOS, DISEÑO DE UN PROTOTIPO”**, ha considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación de Proyecto.

Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

Latacunga, marzo, 2021

Para constancia firman:

Lector 1(Presidente)

Nombre: PhD. Juan Carlos
Chancusig Chisag
CC: 0502275779

Lector 2

Nombre: Ing. Mg. Segundo
Humberto Corrales Beltrán
CC: 0502409287

Lector 3

Nombre: Ing. Mg. Karla Susana Cantuña Flores
CC: 0502305113

CARTA DEL COORDINADOR DEL PROYECTO DE INVESTIGACIÓN



UNIVERSIDAD TÉCNICA DE COTOPAXI

Latacunga, marzo 9 de 2021

CERTIFICACIÓN

Por medio de la presente certifico que el estudiante **Guevara Naranjo Christian Paul** con número de cedula **172299198-9**, desde el periodo académico Marzo 2019 – Agosto 2019 hasta la actualidad, es parte del proyecto de investigación formativo “**Aplicación del modelo Iterativo – Incremental en el desarrollo de herramientas informáticas para instituciones, organizaciones y empresas del entorno educativo, productivo y comercial de la Provincia de Cotopaxi. Proyecto vinculado a los estudiantes de la carrera de Ingeniería en Informática y Sistemas Computacionales**”.

El trabajo del mencionado estudiante se enfocó en el estudio y análisis de metodologías de desarrollo para aplicaciones basadas en microservicios, el mismo está terminado y puede ser presentado y defendido como su proyecto de titulación bajo la modalidad de Proyecto de Investigación.

Es cuanto puedo informar, autorizo al beneficiario hacer uso del presente certificado para los fines legales pertinentes.

DOCENTE RESPONSABLE DEL PROYECTO DE INVESTIGACION FORMATIVA

Ing. MSc. Verónica Tapia Cerda
C.I: 050205369-7

AGRADECIMIENTO

En primer lugar, agradezco a Dios, por brindarme salud y vida para poder cumplir mis sueños, en segundo lugar, a mis padres y hermanos quienes estuvieron conmigo en todo momento apoyándome y dándome fuerzas para salir adelante sin importar las dificultades.

También ofrezco mis más sinceros agradecimientos a todos los docentes que formaron parte de mi vida universitaria, y especialmente a la Ing. Mg. Verónica Tapia por ser una excelente profesional, además de brindarme toda su ayuda durante el desarrollo de este proyecto de investigación, y así poder cumplir uno de mis más grandes sueños.

CHRISTIAN

DEDICATORIA

Este proyecto de investigación se lo dedico a Dios, y a mis queridos papás Jesús y Verónica por saber guiarme por el camino correcto y estar conmigo en todo momento de manera incondicional, a mis hermanos Verito y Kevin por formar parte de mi vida permitiéndome sentirme completo para poder lograr todo lo que me proponga, y por último a mi hermanita Erika por cuidarme desde el cielo todo el tiempo.

CHRISTIAN

ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA.....	II
AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN.....	III
APROBACIÓN DEL TRIBUNAL DE TITULACIÓN.....	IV
CARTA DEL COORDINADOR DEL PROYECTO DE INVESTIGACIÓN.....	V
AGRADECIMIENTO.....	VI
DEDICATORIA.....	VII
ÍNDICE GENERAL.....	VIII
RESUMEN DEL PROYECTO.....	XIII
ABSTRACT.....	XIV
AVAL DE TRADUCCIÓN.....	XV
1. INFORMACIÓN GENERAL.....	1
1.1 Título del proyecto.....	1
1.2 Fecha de inicio.....	1
1.3 Fecha de finalización.....	1
1.4 Lugar de ejecución.....	1
1.5 Facultad que auspicia.....	1
1.6 Carrera que auspicia.....	1
1.7 Proyecto de investigación vinculado.....	1
1.8 Equipo de Trabajo.....	1
1.9 Área de Conocimiento.....	1
1.9.1 Línea de investigación.....	1
1.9.2 Sub líneas de investigación de la Carrera.....	1
3. JUSTIFICACIÓN DEL PROYECTO.....	2
4. BENEFICIARIOS DEL PROYECTO.....	2
5. EL PROBLEMA.....	3

6. OBJETIVOS.....	4
6.1 General.....	4
6.2 Específicos.....	4
7. ACTIVIDADES Y SISTEMA DE TAREAS EN RELACIÓN A LOS OBJETIVOS PLANTEADOS.....	5
8. FUNDAMENTACIÓN CIENTÍFICO TÉCNICA.....	6
8.1 Antecedentes.....	6
8.2 Aplicaciones.....	7
8.2.1 Aplicación Web.....	7
8.3 Modelo Iterativo – Incremental.....	7
8.4 Ingeniería de Software.....	8
8.5 Instrumentos de recolección de datos.....	8
8.5.1 Encuesta.....	8
8.6 Practicas ágiles.....	8
8.6.1 Roles del Equipo.....	8
8.7 Herramientas de desarrollo.....	9
8.7.1 Backend.....	9
8.7.2 Frontend.....	13
8.7. Microservicios.....	14
8.8. Diferencia entre Arquitectura Monolítica y Arquitectura de Microservicios.....	15
8.9. Atributos de calidad en Arquitectura de Microservicios.....	18
8.9.1 Patrones de Microservicios.....	19
8.9.2 Arquitectura Orientada a Microservicios.....	20
8.9.3 Metodología de desarrollo de Microservicios.....	21
8.10 Términos Básicos.....	24
9. PREGUNTAS CIENTÍFICAS O HIPÓTESIS.....	24
9.1 Verificación de la propuesta.....	24

9.2 Juicio de expertos.....	24
9.3 Alfa de Cronbach.....	25
10. METODOLOGÍAS Y DISEÑO EXPERIMENTAL.....	25
10.1. Tipos de Investigación.....	25
10.1.1 Investigación bibliográfica.....	25
10.2 Metodología aplicada al proyecto.....	25
10.2.1 ¿Qué es una metodología de desarrollo?.....	25
10.2.2 Metodologías ágiles.....	26
10.2.3 Modelo iterativo incremental.....	27
10.2.4 Metodología devops.....	27
10.2.5 Metodología MicroIoT.....	28
11. ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS.....	32
11.1 Arquitectura del Prototipo.....	32
11.1.1 Título del Prototipo.....	32
11.1.2 Qué hace el Prototipo.....	32
11.1.3 Objetivos.....	33
11.1.4 Fechas críticas.....	33
11.1.5 Miembros del equipo de desarrollo.....	33
11.1.6 Product backlog.....	34
11.1.7 Prácticas ágiles usadas durante el desarrollo del proyecto.....	34
11.1.8 Tecnologías usadas durante el desarrollo del proyecto.....	35
11.1.9 Conexión de los Microservicios con sus respectivas bases de datos.....	35
11.1.10 Diagrama de bloques.....	36
11.1.11 Modelo de datos.....	36
11.1.12 Release Plan Final.....	37
11.1.13 Detalles de todas la Historias de usuario.....	37
11.1.14 Pruebas realizadas.....	40

11.1.15 Acrónimos	49
12. IMPACTOS (TÉCNICOS, SOCIALES, AMBIENTALES O ECONÓMICOS)	50
12.1 Impacto Técnico	50
12.2 Impacto Social	50
12.3 Impacto Económico	50
13. PRESUPUESTO GENERAL	51
13.1 Gastos directos	51
13.2 Gastos indirectos	51
13.3 Gastos generales	52
13.4 Estimación del costo del software por puntos de historia	52
14. CONCLUSIONES Y RECOMENDACIONES	53
14.1. Conclusiones	53
14.2. Recomendaciones	53
15. BIBLIOGRAFÍA	54
16. ANEXOS	58
16.1 Instrumento utilizado para la validación de la propuesta.	58
16.2 Encuesta	58
16.3 Cartas dirigidas a los expertos	59

ÍNDICE DE TABLAS

Tabla 1. Beneficiarios del proyecto.....	3
Tabla 2. Actividades del sistema	5
Tabla 3. Ventajas y Desventajas - Arquitectura de Microservicios y Arquitectura Monolítica	16
Tabla 4. Tabla comparativa – MicroIoT y DevOps	23
Tabla 5. Rol del equipo – Scrum Master	33
Tabla 6. Rol del equipo – Scrum Team	33

Tabla 7. Sprints a desarrollar	34
Tabla 8. Release Plan Final	37
Tabla 9. Historia de usuario 1 – Crear Usuario (Alumno).....	37
Tabla 10. Historia de usuario 2 – Editar Usuario (Alumno).....	38
Tabla 11. Historia de usuario 3 – Eliminar Usuario (Alumno)	39
Tabla 12. Caso de Prueba CP_001	43
Tabla 13. Caso de Prueba CP_002	44
Tabla 14. Caso de Prueba CP_003	46
Tabla 15. Tabulación de la encuesta, pregunta N° 4.	47
Tabla 16. Tabla estadística del método de Alfa de Cronbach	48
Tabla 17. Rango de confiabilidad.....	49
Tabla 18. Estimación de costos – Gatos Directos	51
Tabla 19. Estimación de costos – Gatos Indirectos.....	51
Tabla 20. Estimación de costos – Gastos Generales.....	52

ÍNDICE DE FIGURAS

Figura 1. Diagrama - Conexión Microservicio - Base de Datos	35
Figura 2. Diagrama de bloques – Módulos del prototipo.....	36
Figura 3. Modelo Relacional	36
Figura 4. Implementación de “Crear Alumno”	40
Figura 5. Implementación de “Editar Alumno”	41
Figura 6. Implementación de “Eliminar Alumno”	42
Figura 7. Tabulación pregunta 4.....	47

RESUMEN DEL PROYECTO

Título:

**ESTUDIO METODOLÓGICO PARA EL DESARROLLO DE MICROSERVICIOS,
DISEÑO DE UN PROTOTIPO.**

Autor:

Guevara Naranjo Christian Paúl

El presente proyecto se basó en una revisión bibliográfica, que detalla las metodologías para el desarrollo y creación de Microservicios, mediante la utilización de prácticas ágiles como son el desarrollo Iterativo – Incremental, historias de usuario, entre otras. La investigación bibliográfica permitió resaltar el cambio que ha tenido la industria del software en cuestión de otras formas de implementación, realizando una comparación entre arquitecturas tradicionales más utilizadas y las nuevas basadas en microservicios, el objetivo principal fue analizar el proceso metodológico que sigue para el desarrollo de microservicios y proponer una estrategia donde se incluyan atributos de calidad en este proceso de desarrollo. A demás, ante la falta de conocimiento en nuestro medio y al ser una tendencia portadora de futuro en la industria del desarrollo de software, otro de los resultados importantes de este estudio fue, el diseño y elaboración de un prototipo basado en una arquitectura de microservicios, que cuenta con un conjunto de características y funcionalidades, que sirven como punto de partida para futuros investigadores y desarrolladores sobre todo de la carrera de Ingeniería en Sistemas de Información de la Universidad Técnica de Cotopaxi, y como herramienta demostrativa del comportamiento, comunicación y trabajo de los microservicios en una aplicación en ejecución, además del potencial que aportan las herramientas tecnológicas usadas para la creación de este nuevo tipo de aplicaciones.

Palabras clave: microservicios, desarrollo de software, arquitecturas, prototipo, prácticas ágiles.

TECHNICAL UNIVERSITY OF COTOPAXI

FACULTY OF SCIENCE AND ENGINEERING APPLIED

**THEME: METHODOLOGICAL STUDY FOR THE DEVELOPMENT OF
MICROSERVICES, DESIGN OF A PROTOTYPE.**

ABSTRACT

This project was based on a bibliographic review, which details the methodologies for the development and creation of microservices, through the use of agile practices such as iterative development - Incremental, user stories, among others. The bibliographic research found, it permitted to highlight the change that the software industry has had in question from other forms of implementation, making a comparison between the most used traditional architectures and the new ones based on microservices, the main objective was to analyze the methodological process that follows for the development micro services and getting a strategy that includes quality attributes in this development process. In addition, given the lack of knowledge in our environment and is being a trend that carries the future in the software development industry, another of the important results of this study was, the designing and elaboration of a prototype based on a microservices architecture, which has a set of characteristics and functionalities, which serve as a starting point for future researchers and designers, especially in the Information Systems Engineering career of the Technical University of Cotopaxi, and as a demonstrative tool of behavior, communication and work. of micro services in a performance application, in addition to the potential provided by the technological tools used to create this new type of application.

Keywords: microservices, software development, architectures, prototype, agile practices.

AVAL DE TRADUCCIÓN



Universidad
Técnica de
Cotopaxi

CENTRO DE IDIOMAS

AVAL DE TRADUCCIÓN

En calidad de Docente del Centro de Idiomas de la Universidad Técnica de Cotopaxi; en forma legal **CERTIFICO** que: La traducción del resumen del proyecto de investigación al Idioma Inglés presentado por el estudiante **GUEVARA NARANJO CHRISTIAN PAÚL** de la Carrera de **INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES**, de la **FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS**, cuyo título versa **"ESTUDIO METODOLÓGICO PARA EL DESARROLLO DE MICROSERVICIOS, DISEÑO DE UN PROTOTIPO"**, lo realizó bajo mi supervisión y cumple con una correcta estructura gramatical del idioma.

Es todo cuanto puedo certificar en honor a la verdad y autorizo al peticionario hacer uso del presente certificado de la manera ética que estimare conveniente.

Latacunga, marzo 2021

Atentamente,

Lic. Marcelo Pacheco
DOCENTE CENTRO DE IDIOMAS
C.C. 0502617350

1803027935
VICTOR HUGO
ROMERO
GARCIA

firmado digitalmente
por 1803027935
VICTOR HUGO
ROMERO GARCIA
Fecha: 2021.03.09
12:59:32 -05'00'

1. INFORMACIÓN GENERAL

1.1 Título del proyecto

Estudio metodológico para el desarrollo de Microservicios, diseño de un prototipo.

1.2 Fecha de inicio

5 de noviembre del 2020

1.3 Fecha de finalización

18 de febrero del 2021

1.4 Lugar de ejecución

Universidad Técnica de Cotopaxi

1.5 Facultad que auspicia

Facultad de Ciencias de la Ingeniería y Aplicadas

1.6 Carrera que auspicia

Ingeniería en Informática y Sistemas Computacionales

1.7 Proyecto de investigación vinculado

Aplicación del modelo Iterativo - Incremental en el desarrollo de Herramientas Informáticas para instituciones, organizaciones y empresas del entorno educativo, productivo y comercial de la Provincia de Cotopaxi. Proyecto vinculado a los estudiantes de la carrera de Ingeniería en Informática y Sistemas Computacionales.

1.8 Equipo de Trabajo

Tutor: MSC. Verónica del Consuelo Tapia Cerda

Estudiante: Christian Paul Guevara Naranjo

1.9 Área de Conocimiento

Ciencias

1.9.1 Línea de investigación

Tecnologías de la Información y Comunicación (TICS)

1.9.2 Sub líneas de investigación de la Carrera

Ciencias informáticas para la modelación de Sistemas de Información a través del desarrollo de software.

3. JUSTIFICACIÓN DEL PROYECTO

“En la actualidad, a nivel empresarial y tanto en el sector privado como público se realiza desarrollo de software para suplir las necesidades de automatización de procesos internos, este desarrollo ha seguido las tendencias impuestas por la plataforma, lenguaje de programación o por la experiencia del área de desarrollo, lo cual deviene en la implantación de sistemas de construcción tradicional o monolítico.”(Maya, E., y López, 2017) Y ya que en el pasado para el diseño de aplicaciones se ocupaba la arquitectura monolítica, la cual consistía básicamente en que todos los componentes o elementos que tenían la posibilidad de implementarse estaban estructurados en un solo campo, con la aplicación de este método genera desventajas porque cuanto más grande es una aplicación se vuelve una tarea mucho más difícil el tener la capacidad de solucionar de manera rápida y efectiva problemas que se puedan presentar o aparecer, pero es todo lo contrario si estas aplicaciones o sistemas se diseñan utilizando microservicios porque tiene la capacidad de resolver dichos problemas, ya que con los microservicios a las aplicaciones se las puede desintegrar en varias partes de manera sencilla, por motivo de que son elementos independientes que trabajan en conjunto para lograr una tarea mucho más grande, así el microservicio cumple con su objetivo el cual es desarrollar y perfeccionar sistemas de software de calidad con mayor rapidez de actuación y mejoramiento de procesos a comparación del frecuente modelo habitualmente utilizado, tomando en cuenta todas estas ventajas, que representa utilizar esta arquitectura, se pretende diseñar un prototipo para demostrar el funcionamiento de los microservicios con todas sus características y con esto contribuir al desarrollo de sistemas y aplicaciones que se pretenda desarrollar, por parte de futuros estudiantes de la carrera de Ingeniería en Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi.

4. BENEFICIARIOS DEL PROYECTO

Al generar este proyecto de investigación es evidente que van a existir beneficiarios tanto directos como indirectos, mismo que vienen a ser específicamente los futuros estudiantes de la carrera de Ingeniería en informática y sistemas computacionales de la Universidad Técnica de Cotopaxi, que lleguen a conocer el tema de microservicios, sea de su agrado y elijan adentrarse más a fondo, les será de gran ayuda el prototipo que se desarrolla como muestra del funcionamiento, relación, comunicación y comportamiento que tienen los microservicios al trabajar independientemente pero entre sí a la vez para lograr cumplir con un objetivo más grande, además de los estudiantes de la Universidad Técnica de Cotopaxi toda persona, ya sea

desarrollador, analista de sistemas o un simple apasionado de la informática le será de gran ayuda igualmente para lograr un mayor entendimiento del tema.

Tabla 1. Beneficiarios del proyecto

BENEFICIARIOS	
BENEFICIARIOS DIRECTOS	Estudiantes de la carrera de Ingeniería en Informática y Sistemas Computacionales de la Universidad Técnica de Cotopaxi.
BENEFICIARIOS INDIRECTOS	Usuarios de los sistemas elaborados con microservicios.

Fuente: Christian Guevara (Investigador)

5. EL PROBLEMA

En los inicios del desarrollo de aplicaciones, se generaban inconvenientes al realizar cambios en la estructura de un proyecto de desarrollo de software por más pequeños que sean, para poder solucionar estos errores se necesitaba una intervención y manipulación completa de todo el sistema de software o aplicación que evidentemente retrasaba a varios grupos de desarrolladores, el proceso que presenta estas desventajas es el enfoque monolítico tradicional, ya que todos sus componentes desarrollados son implementados en una sola estructura, y si por algún motivo existía algún error, obligatoriamente se debía analizar todo el sistema por completo, revisarlo, identificarlo y por último arreglar dicho problema, este enfoque es recomendado para aplicaciones pequeñas con procesos que no representen una cantidad importante de desarrollo, que al encontrarse con algún tipo de problema sea sencillo de identificar y solucionar por el mismo hecho de no ser tan extenso, pero en ningún concepto es recomendable para empresas grandes las cuales trabajan con aplicaciones que manejan procesos que conlleven una gran cantidad de intercambio de información. “Las aplicaciones que utilizan este estilo arquitectónico tienden a ser diseñadas para tolerar fallas y errores de manera resiliente”(Pareja Valerio & Burgos Robles, 2019) Actualmente grandes empresas se han visto en la necesidad de buscar alternativas para poder satisfacer sus necesidades, cambiando de arquitectura para mejorar sus procesos al migrar sus datos a la arquitectura de microservicios

corrigiendo así problemas de estimación de esfuerzos, costos y tiempo. “Los microservicios pueden ser reemplazados fácilmente, por ende, recuperarse de un fallo es relativamente sencillo. Un nuevo microservicio no necesita utilizar parte de la base de código ni las tecnologías del antiguo microservicio. Pequeños microservicios facilitan además su actualización y cuando la decisión de una tecnología o enfoque se limita a un microservicio, este puede ser completamente reescrito si es necesario.”(Arboleda Cola Carlos Augusto, 2017)

En la Universidad Técnica de Cotopaxi se evidencia la falta de información acerca del tema de microservicios y de un prototipo que sirva como ejemplo demostrativo para conocer el funcionamiento del mismo, para futuros estudiantes que opten por elaborar investigaciones posteriores del tema, y al no contar con un sistema que sirva como punto de partida, que demuestre los pasos y herramientas tecnológicas con las cuales se pueda profundizar y adquirir nuevos conocimientos de manera más rápida y eficaz.

¿Cómo se puede apoyar a la innovación del desarrollo de software orientado a microservicios en la carrera de Ingeniería en Sistemas de Información de la Universidad Técnica de Cotopaxi?

6. OBJETIVOS

6.1 General

Analizar el proceso metodológico para el desarrollo de microservicios, a través de una revisión bibliográfica que permita proponer una estrategia para su implementación.

6.2 Específicos

1. Realizar un estudio bibliográfico sobre metodologías de desarrollo de Microservicios como una tendencia portadora de futuro en la industria de software.
2. Proponer una estrategia de desarrollo para Microservicios considerando atributos de calidad en el proceso.
3. Diseñar un prototipo de Microservicios aplicando la estrategia propuesta.

7. ACTIVIDADES Y SISTEMA DE TAREAS EN RELACIÓN A LOS OBJETIVOS PLANTEADOS

Tabla 2. Actividades del sistema

Objetivos	Actividad	Resultado de la actividad	Descripción de la actividad (técnicas e instrumentos)
Realizar un estudio bibliográfico sobre metodologías de desarrollo de Microservicios como una tendencia portadora de futuro en la industria de software.	<ol style="list-style-type: none"> 1. Realizar la investigación de fuentes bibliográficas confiables, primarias y secundarias sobre microservicios. 2. Investigar la evolución que ha tenido los microservicios a lo largo del tiempo. 	<ul style="list-style-type: none"> - Seleccionar metodologías que tengan relación directa con el desarrollo de Microservicios, las cuales sirvan para diseñar un prototipo. 	<ul style="list-style-type: none"> - Utilizar como técnica Juicio de expertos y un cuestionario de validación como instrumento de recolección de datos para verificar la confiabilidad del proyecto.
Proponer una estrategia de desarrollo para Microservicios considerando atributos de calidad en el proceso.	<ol style="list-style-type: none"> 3. Analizar todos los datos recopilados de cada tema. 4. Investigar acerca de los atributos de calidad que pueden adaptarse para el desarrollo de microservicios. 	<ul style="list-style-type: none"> - Identificar los atributos de calidad y estrategias necesarios desarrollar microservicios. 	<ul style="list-style-type: none"> - Utilizar la técnica de la revisión bibliográfica para determinar atributos de calidad, empleando el instrumento de catálogo de referencias.
Diseñar un prototipo de Microservicios aplicando la estrategia propuesta.	<ol style="list-style-type: none"> 5. Desarrollar el prototipo de microservicios utilizando la estrategia propuesta para demostrar los resultados obtenidos mediante la investigación bibliográfica. 	<ul style="list-style-type: none"> - Establecer funcionalidades, las cuales estructuren el prototipo en su totalidad. 	<ul style="list-style-type: none"> - Mediante la revisión bibliográfica establecer las herramientas tecnológicas adecuadas para el diseño del prototipo, utilizando como instrumento el catálogo de referencias.

Fuente: Christian Guevara (Investigador)

8. FUNDAMENTACIÓN CIENTÍFICO TÉCNICA

8.1 Antecedentes

En (2019) se presentó una propuesta de un sistema móvil para mejorar la gestión y control académico en la Escuela Profesional de Obstetricia de la Universidad Nacional Mayor de San Marcos.

En esta propuesta desarrollaron un sistema utilizando metodologías ágiles con el objetivo de mejorar la gestión y el control académico que tenían en su institución, y que además de esta metodología optaron por utilizar la arquitectura de microservicios como herramienta para optimizar todos sus procesos con lo que pretenden satisfacer varias necesidades y carencias como mejorar específicamente la parte de integración y rendimiento , aprovechando al máximo las ventajas que ofrecen los microservicios y sus características principales como flexibilidad, total independencia, y acoplamiento, producto del agrupamiento de esta arquitectura y una metodología ágil tuvieron como resultado un proyecto muy bien organizado, donde el usuario comprueba que todas las necesidades surgidas en un inicio fueron resueltas en su totalidad y en el tiempo establecido.

Tomando referencia del proyecto anteriormente citado se puede notar de manera clara que ya teniendo un sistema implementado, el objetivo principal y que se convierte en una necesidad es mejorar a toda costa los procesos que este realiza, optando por la arquitectura de microservicios como herramienta para generar un mejor rendimiento, lo cual lo transforma en un sistema eficaz y seguro, el cual posea la posibilidad de trabajar de la manera más apropiada y que a su vez cumpla con todos los requerimientos establecidos.

En (2020) se desarrolló un E-commerce basado en microservicios para el proceso de ventas en la empresa COMATPERU S.A.C.

El resultado final que se obtuvo producto del desarrollo de este ecommerce utilizando arquitectura de microservicios como herramienta principal para obtener mayores ventas y ganancias para la empresa fueron porcentajes altos y favorables a partir de su implementación, ya que “el E-commerce aumento el porcentaje de crecimiento de ventas en el lapso de tiempo determinado en 5.45%. Presentando en un inicio el valor de 16.45% y posterior a ello un valor de 21.90%. Por lo cual se afirma que el E-commerce aumento el porcentaje de crecimiento de ventas en el tiempo determinado en el proceso de ventas.”(Zafra Ramirez, 2020) evidenciando así que desarrollar sistemas y aplicaciones utilizando microservicios mejora notablemente el

rendimiento, además de sentar precedentes e información sobre esta arquitectura que en los últimos años y actualmente ha sido un gran impacto en la industria tecnológica.

“Desarrollo ágil del nuevo sistema institucional basado en una nueva arquitectura orientada en microservicios.”(Rivadeneira, 2019)

En el tema anteriormente mencionado se estableció como objetivo principal cambiar la arquitectura tradicional monolítica con la que contaban en un principio, por la de microservicios con el fin de demostrar que, con la utilización de esta arquitectura, tendrían la posibilidad de obtener una mejoría en el rendimiento, como es la agilidad en cuanto a procesos, la cual es una necesidad que se desea satisfacer en cada uno de sus módulos implementados.

8.2 Aplicaciones

8.2.1 Aplicación Web

“Son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores Web en la que se confía la ejecución al navegador.”(Hernández et al., 2011)

Las aplicaciones web son una especie de software que utilizan navegadores como herramientas de interpretación y ejecución, es necesario mencionar que un requisito fundamental para poder denominarse como una aplicación web es tener un servidor que esté relacionado mediante internet, pero utilizando obligatoriamente un navegador, así y solo así es que podrá conectarse y posteriormente ejecutarse.

8.3 Modelo Iterativo – Incremental

“Permite la entrega de versiones parciales a medida que se va construyendo el producto final. Es decir, a medida que cada incremento definido llega a su etapa de operación y mantenimiento. Cada versión emitida incorpora a los anteriores incrementos las funcionalidades y requisitos que fueron analizados como necesarios.”(DÍAZ, n.d.)

Con lo mencionado en la anterior cita se puede entender que el modelo iterativo – incremental esta particionado e inicia por etapas o iteraciones teniendo en cuenta los requerimientos que se han establecido previamente por parte del solicitante o usuario, en primer lugar se atraviesa por etapas como análisis, diseño, código y pruebas, con las etapas mencionadas se tiene la posibilidad de evaluar el cumplimiento y funcionalidad de cada una de las tareas registradas,

con el fin de culminar el proyecto establecido desde un inicio y finalmente poder entregar un producto final de calidad.

8.4 Ingeniería de Software

“La ingeniería de software está compuesta por una serie de modelos que abarcan los métodos, las herramientas y los procedimientos. Estos modelos se denominan frecuentemente paradigmas de la ingeniería del software y la elección de un paradigma se realiza básicamente de acuerdo al tipo del proyecto y de la aplicación, los controles y las entregas a realizar.”(Cataldi, Z, Lage, F, Pessacq, 1997)

Esta ingeniería tiene completa relación con la elaboración y desarrollo de software, que a su vez es denominado como ciclo de vida de software, el cual permite que el proceso que se lleve para desarrollar software este apto para cualquier aplicación, utilizando siempre el análisis, diseño, codificación, implementación y mantenimiento del mismo, este conjunto de etapas vitales para un desarrollador al momento de trabajar con prácticas ágiles en el mundo del desarrollo de software.

8.5 Instrumentos de recolección de datos

8.5.1 Encuesta

La encuesta es uno de los instrumentos de recolección y análisis de datos más utilizados por los investigadores, ya que posee una recopilación de técnicas de investigación social mediante la cual se convierte en una actividad en la que todos alguna vez llegaremos a utilizar y participar, se usa básicamente para obtener todo tipo de información mediante un cuestionario que se aplica a un grupo de personas, tendiendo la posibilidad de abordar cualquier tipo de tema sin ningún tipo de restricción o inconveniente.

8.6 Practicas ágiles

8.6.1 Roles del Equipo

8.6.1.1 Scrum master

“El Scrum Master se asegura de que todo el mundo (incluidos el Dueño de Producto y los gerentes) comprenden los principios y las prácticas de Scrum, y ayudan a guiar a la organización a través de los habitualmente difíciles cambios que son necesarios para lograr el éxito en el desarrollo Ágil.” (Deemer et al., 2012)

Es la persona que lidera y apoya a un determinado equipo de trabajo que está a cargo de un respectivo proyecto de desarrollo, teniendo la capacidad de poder guiarlos y verificarlos

constantemente para cumplir todos sus objetivos planteados, tomando la responsabilidad completa y afrontando cualquier dificultad que le pueda surgir al equipo de trabajo.

8.6.1.2 Scrum Team

Se denomina Scrum Team al equipo que se le asigne la tarea de entregar el producto establecido en el proyecto a desarrollar de la manera iterativa o incremental, el equipo anteriormente mencionado debe poseer y cumplir obligatoriamente con ciertas características, como tener los suficientes conocimientos para tener la capacidad profesional de poder llevar a cabo sus tareas y obligaciones.

8.6.1.3 Product Backlog

“El Product Backlog, es conocido como la lista de tareas a desarrollar definidas y especificadas en la primera reunión.”

Básicamente la función de esta práctica ágil es una lista donde se detallan todos los requerimientos que se plasmaron mediante un previo estudio al principio del proyecto que se va a desarrollar, cabe mencionar que con el pasar del tiempo va a seguir evolucionando con su respectivo control obviamente, según la constancia del avance de entrega del producto.

8.6.1.4 Historias de usuario

“Describen funcionalidades que dan solución a necesidades o problemas del cliente o del usuario, representan los "qués" a construir y se escriben en forma de historia con una o dos frases utilizando el lenguaje común del usuario.”(Menzinsky, Alexander, López, Gertrudis, Palacio, 2018)

Las historias de usuario son básicamente parte de un conjunto de prácticas ágiles cuya función principal es detallar de manera clara y concisa los requerimientos de un determinado proyecto que se establecieron previamente, con el fin de lograr un mejor entendimiento.

8.7 Herramientas de desarrollo

8.7.1 Backend

8.7.1.1 Eureka

“Este servicio se comporta como un servidor, centralizando la información de los servicios disponibles, permitiendo localizarlos y proveer sus endpoints cuando sea necesario. Para su correcto funcionamiento, cada vez que se arranque un microservicio debe comunicarse con el

servidor Eureka, notificando que está disponible y enviando su dirección y sus meta datos”.(Saavedra, 2019)

Con todo lo anteriormente mencionado se puede entender que Eureka es el servidor que se debe utilizar si se va a trabajar con servicios y microservicios especialmente ya que estos se pueden registrar y localizar sin ningún tipo de problema, ya que este servidor tiene la opción de poder configurar para la conexión y comunicación entre microservicios por medio de clústeres que básicamente son procesos programados mediante las cuales se intercambia todo tipo de información con su debido control y verificación, lo que hace de este servidor y de la utilización de microservicios tan especial es precisamente la tolerancia a fallos, es decir tener la capacidad de reconocer y solucionar problemas oportunamente, ya que como en todo sistema van a enfrentar cualquier tipo de inconvenientes, y que se pueden resolver de manera rápida y eficaz mediante la utilización de este servidor y la arquitectura de microservicios, cabe mencionar un dato importante a resaltar, el cual es que este servidor fue desarrollado por Netflix, cabe mencionar que esta grande empresa ha dado mantenimiento y soporte técnico los últimos años a esta arquitectura ayudándola constantemente a evolucionar en el ámbito tecnológico, ya que en su composición precisamente alberga más de 500 microservicios trabajando en conjunto.

8.7.1.2 Framework Spring Cloud

“Spring Cloud ofrece un modelo de programación en un entorno Java, que está construido sobre Spring Boot.”.(Aguirre, 2018)

Este framework al tener relación directa con el lenguaje de programación Java y usarlo como su entorno predeterminado, facilita y proporciona puertos y configuraciones las cuales hace del manejo y desarrollo de microservicios una tarea sencilla de realizar, ya que en estos se usan atributos de calidad y patrones, los cuales son características esenciales de esta arquitectura, gracias a todas estas ventajas que ofrece este framework es que se mejora el rendimiento de las aplicaciones construidas a base de este, y al mismo tiempo reduce notablemente y en gran proporción el tiempo de desarrollo, por lo que esto representa menor gasto, en cuestión de estimación de costos, tiempo y esfuerzo.

Se tiene que resaltar también a los proyectos, aplicaciones, y sistemas que son construidos a base de este framework ya que con Spring Cloud tienen la posibilidad no menorar el tiempo de ejecución es decir ponerlos en marcha en muy poco tiempo, con todos sus objetivos cumplidos.

8.7.1.3 Base de datos

“Una base de datos es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos. Cada base de datos ha sido diseñada para satisfacer los requisitos de información de una empresa u otro tipo de organización, como, por ejemplo, una universidad o un hospital.”(Marqués, 2009)

En sí, una base de datos es información estructurada y bien organizada que se almacenan en un sistema, principalmente permite guardar datos, realizar búsquedas, ingresar nuevos datos, todos estos procesos se realizan de manera sencilla y eficiente desde una computadora, los datos ingresados son almacenados, registrados y organizados mediante tablas previamente estructuradas e implementadas por defecto.

8.7.1.4 PostgreSQL

“PostgreSQL es un gestor de bases de datos orientadas a objetos (SGBDOO o ORDBMS en sus siglas en inglés) muy conocido y usado en entornos de software libre porque cumple los estándares SQL92 y SQL99, y también por el conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo o a un mejor nivel que muchos SGBD comerciales.”(Ginestà, Marc Gibert, 2012)

PostgreSQL es un gestor de base de datos, una de sus características más importantes es que es relacional y orientado a objetos, cuenta la facilidad de utilizar código abierto, este gestor de base de datos se mantiene gracias a la intervención de varias entidades que valen por su continuo mantenimiento, como desarrolladores y diversas organizaciones, las cuales aportan con soporte técnico sin ningún tipo de retribución o beneficio personal.

PostgreSQL cuenta con muchas ventajas y desventajas, pero si se lo aprovecha al máximo, se lo puede usar principalmente para almacenar y procesar una incomparable cantidad de datos e información (DWH), se lo puede utilizar también en servicios web para todo tipo de aplicaciones, existen registros y evidencias de un sin número de plataformas que usan este gestor de base de datos como fuente de servicio, como garantía de esto tenemos el caso de Skype que lo usa como servicio para sus diferentes departamentos de telefonía y más, al ser una herramienta Open Source significa un gran aporte para quien opte y tenga la capacidad de utilizarlo.

8.7.1.5 MariaDB

“MariaDB está basado en MySQL con más funcionalidades y mejor rendimiento. MariaDB es un fork de MySQL que nace bajo la licencia GPL v2.”(SANTIAGO ALEJANDRO GARCÍA VELA., 2016)

MariaDB es un sistema de gestión de base de datos más conocidos a nivel mundial que principalmente nace o se desprende de MSQL y que como ventaja importante tiene una total compatibilidad, este gestor tiene funciones principales como almacenar y extraer datos de una base de datos externa ya que cuenta con licencia GPL, aunque por falta de información en la mayoría de ocasiones se compara a MariaDB y MSQL son sistemas completamente diferentes y que en la mayoría de los casos Mariadb ofrece mejor rendimiento al ser un sistema Open Source, es así que este motor de base de datos ha sido prácticamente con el tiempo y el pasar de los años se ha convertido en el reemplazo de MSQL ya que gracias a ser en su totalidad compatible no existe ningún problema si se quiere migrar cualquier tipo de sistema o dato de un gestor de base de datos a otro.

8.7.1.6 MongoDB

“MongoDB es una base de datos orientada a documentos agrupados en colecciones y permite expresar las consultas en diferentes lenguajes, por ejemplo, PHP, Node.js, Python, Ruby o Java.”(Kenneth Calvo, Johan Durán, Esteban Quirós, 2017)

MongoDB a diferencia de las demás bases de datos en lugar de almacenar datos e información en registros con tablas, almacena estos datos en tipo de estructuras que se le pueden denominar documentos, los cuales son registrados gracias a archivos Bson, mismos que son la representación de estructura Json pero convertidos a binario, al ser una base de datos no relacional, no tiene conflictos ni entre relaciones, ni entre llaves, esto lo convierte en una base de datos mucho más flexible a comparación de las bases de datos SQL o relacionales, y como MongoDB al ser no relacional, no maneja SQL, por ende no tiene una nomenclatura escrita para sí mismo, esto provoca que no maneje relaciones, ni claves foráneas, tampoco maneja tablas sino en su lugar colecciones de documentos, como se menciona anteriormente.

“MongoDB es una base de datos NoSQL de tipo documental de código abierto.”(Seneque, 2014)

8.7.2 Frontend

8.7.2.1 JavaScript

“Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.”(Pérez, 2019)

“JavaScript fue desarrollado por Netscape, a partir del lenguaje Java, el cual sigue una filosofía similar, aunque va más allá. Java es un lenguaje de programación por sí mismo, como lo puedan ser C, Pascal o Visual Basic. Esto quiere decir, que se puede ejecutar un programa Java fuera de un navegador. Se hablará más sobre Java en posteriores temas.”(Navarrete, 2006)

Para entenderlo mejor JavaScript es un lenguaje de programación como muchos otros pero con características que lo hacen único, es un lenguaje diferente y especial ya que es un lenguaje de comandos que te permite plasmar tus ideas directamente a una aplicación web, actualmente es el más utilizado en el mundo de la programación, ya que es un lenguaje interpretado que no tiene la necesidad de ser compilado, HTML y CSS son tecnologías con las cuales uniéndolas son el complemento ideal para este lenguaje, con este conjunto se puede crear cualquier tipo de aplicación web sin ningún tipo de inconveniente o limitación.

8.7.2.2 Angular

“Angular es un Framework desarrollado por la empresa Google, que permite la creación de aplicaciones web, especialmente un cliente Front-End esto quiere decir que es un sitio web que interactúa con el cliente. Se encuentra estructurada por el lenguaje HTML, JavaScript y TypeScript. Es una plataforma para la creación de aplicaciones web orientado a proyectos de mediana a gran escala.”(Gudiño, 2018)

“Angular comenzó en el 2010 bajo el nombre de AngularJS. AngularJS proporcionaba controladores y directivas para dividir el código en JavaScript para poder manejar la interfaz gráfica de usuario y las peticiones al servidor por aparte. Para la segunda versión de AngularJS, el equipo desarrollador decidió cambiar la nomenclatura del framework a simplemente Angular.”(Ramon, 2020)

Básicamente Angular es un framework desarrollado a base de código nativo de JavaScript, con este framework se puede desarrollar aplicaciones web con la facilidad de desarrollar en una sola página, una de sus ventajas es que, si se necesita actualizar una parte determinada de la

aplicación que se esté creando, se puede hacer justamente eso, evitando que se actualice toda la aplicación como sucede con otros frameworks y lenguajes de programación lo que un determinado tiempo puede convertirse en un gran problema, pero con Angular al ser su código y programación reactiva cuando se realizan cambios se visualizan automáticamente sin necesidad de actualizar toda la página.

8.7. Microservicios

“Un microservicio se define como: "un repositorio de código pequeño, autónomo y desplegado independientemente de los demás". Se desarrollan y diseñan en base a una capacidad de negocio concreta y pueden ser vistos como una aplicación independiente del resto de microservicios, aun cuando coexistan para la implementación del sistema completo.”(Verdier & Rodriguez, 2020)

Los microservicios tienen características importantes como las siguientes:

- Trabajan en base a una tarea específica.
- Siempre se les asigna una funcionalidad en especial.
- Se entiende claramente los límites entre los diferentes servicios, con esto se tienen autonomía refiriéndose a la implementación, lo que significa mayor cohesión y menor acoplamiento.
- Logran ser independientes al conseguir comunicarse mediante interfaces y no hacer visible su implementación.
- Para mantener un servicio se asignan a un pequeño grupo de desarrolladores.

Cada microservicio es implementado de manera única, siempre diferenciándose de los demás, lo que causa que se logren implementar en paralelo, en consecuencia, su mantenimiento es sencillo, esta arquitectura permite independencia y autonomía en los grupos de desarrollo, por esto se forman grupos específicos para desarrollar microservicios, logrando considerablemente rapidez en el desarrollo.

En cuanto a tecnologías cuando se asignan equipos definidos para trabajar en microservicios se les da la libertad de escoger equipos, es decir no se nos obliga a trabajar con tecnologías que tengan relación una con otra, son libres en lo que respecta a la selección de lenguajes de programación y frameworks, incluso queda a su total elección la utilización de una o más bases de datos, tomando en cuenta previamente ciertas necesidades como es la capacidad de que se

adapte al servicio y funcionalidad requerido. Independientemente del servicio, se busca el hardware que sea apto a para implementarse con sus requerimientos referentes a recursos.

“En un mundo dinámico de desarrollo de software, los estilos arquitectónicos están en continua evolución, adaptándose a las nuevas tecnologías y tendencias, la arquitectura de microservicio (MSA) se adoptó por los profesionales de la industria debido a sus ventajas en comparación con la arquitectura monolítica. Aunque MSA se construye sobre los conceptos centrales de la Arquitectura Orientada a Servicios (SOA), busca una granularidad más fina, con límites más estrictos. Debido a la lógica de los costes, numerosas empresas optan por migrar el estilo monolítico en lugar de desarrollar desde cero. Recientemente las herramientas de descomposición semiautomáticas ayudan al proceso de migración, sin embargo, todavía falta una parte crucial como la validación.”(Cojocarú et al., 2019)

Los microservicios, así como tienen ventajas cuentan con ciertas desventajas por motivo de que no se ha encontrado algún algoritmo específico para separar un sistema en varios servicios, aquí interviene la etapa de descomposición que por cierto en la mayoría de ocasiones suele ser muy complicada en cuestión de cumplir este proceso, ya que con la realización de un mal diseño podría provocar que los beneficios de esta arquitectura no sean aprovechados en su totalidad.

8.8. Diferencia entre Arquitectura Monolítica y Arquitectura de Microservicios

“Si bien se entiende un servicio como una pieza de funcionalidad del sistema, que busca optimizar el aplicativo para la máxima utilidad del consumidor, en donde idealmente cada servicio es diferente ya que se optimiza de forma independiente para los casos de uso de sus clientes.”(Arboleda Cola Carlos Augusto, 2017)

“Los microservicios son una entidad separada, deben ser desplegados como un servicio aislado en infraestructuras que se enfocan en ofrecer plataformas como servicio (PAAS, por sus siglas en inglés) o deben estar en su propio sistema. Todas las comunicaciones entre servicios son llamados de red para reforzar la separación y evitar los peligros del estrecho acoplamiento.” (Alfonso & Contreras, 2018)

Tabla 3. Ventajas y Desventajas - Arquitectura de Microservicios y Arquitectura Monolítica

<p><i>Arquitectura de Microservicios</i></p>	<p style="text-align: center;">Ventajas</p> <ul style="list-style-type: none">• Los microservicios le brindan al desarrollador de software la facilidad de utilizar varias tecnologías y lenguajes de programación.• Cada microservicio trabaja de manera independiente por lo que produce mayor agilidad en los procesos de una aplicación.• Realizar mantenimiento a un microservicio resulta sencillo, gracias a la manera organizada en la que se desarrolla.• Las funcionalidades en cuestión de microservicios se pueden reutilizar en aplicaciones de terceros según la necesidad, no es necesario programarlos de nuevo.• Un sistema o aplicación desarrollado con esta arquitectura trabaja de manera eficaz y sencilla, gracias al trabajo independiente que realiza cada microservicio, esto se ve reflejado en su rendimiento.• Al utilizar microservicios como arquitectura de una aplicación facilita considerablemente la gestión de funcionalidades ya que, al conectarse a más de una base de datos, la información pasa de una manera mucho más rápida. <p style="text-align: center;">Desventajas</p> <ul style="list-style-type: none">• Para desarrollar microservicios se necesitan desarrolladores experimentados en el tema.• Implementar microservicios puede representar gastos significativos.
--	---

Arquitectura Monolítica

Ventajas

- Todos sus componentes tienen una sola implementación.
- El desarrollo de software con esta arquitectura representa un costo directo con respecto a su implementación.
- Los componentes al estar integrados en una sola estructura es sencillo realizar pruebas de rendimiento.

Desventajas

- Al presentarse un error en la aplicación o sistema es difícil encontrar la raíz del problema, producto del acoplamiento de todos sus componentes.
- En grandes aplicaciones, a causa de cómo funciona esta arquitectura, y con un masivo procesamiento de datos pueden representar un retraso en el rendimiento.
- Con esta arquitectura tradicional, toda la información que pasa por los servicios es alojada en una sola base de datos, por lo que a largo plazo puede generar problemas de seguridad.

Fuente: Christian Guevara (Investigador)

8.9. Atributos de calidad en Arquitectura de Microservicios

“Cuando se busca dividir una aplicación grande en partes, a menudo la administración se enfoca en la capa de tecnología, lo que lleva a equipos de IU, equipos de lógica del lado del servidor y equipos de bases de datos. Cuando los equipos se separan a lo largo de estas líneas, incluso los cambios simples pueden llevar a que un proyecto entre equipos requiera tiempo y aprobación presupuestaria.”(Lewis & Fowler, n.d.)

“Finalmente, los principales atributos de calidad relacionados con la arquitectura son escalabilidad, independencia y mantenibilidad, sin embargo, esto contribuye solo a la arquitectura general y los patrones no son temas de estudio”(Valdivia et al., 2019)

En el campo de la arquitectura, los patrones de microservicios también son mencionados e intervenidos, con esto se obtuvo una lista con varios patrones en los cuales se nombran principalmente ventajas y desventajas, lo bueno y lo malo, tomando en cuenta la manera lógica a la coordinación y comunicación entre microservicios por parte de diversas puertas de enlace API. Además del enfoque que debe tener por parte del cliente y el servidor, se usan también estrategias refiriéndose a la implementación.

“Las direcciones de interfaz reales (URL) de los microservicios pueden estar ocultos, mediante inspección de inyección o validación de entrada (tipos de contenido, métodos HTTP) se pueden realizar pruebas equivalentemente para cada servicio, pero por la autorización del proceso este servicio requiere también una base de datos que contenga datos para todos los servicios, lo que produce vulnerabilidades.”(Müssig et al., 2017)

Si hablamos de granularidad estamos hablando de uno de los atributos de calidad más importantes de un microservicio, refiriéndose este atributo básicamente al tamaño con el que se desarrollan los microservicios en un principio, dicho tamaño se puede medir mediante la utilización de líneas de código, que básicamente es un tipo de regla establecida por la cantidad de líneas de código creadas o generadas por los desarrolladores en el proyecto o aplicación en que se esté trabajando. La utilidad principal que se le da a este tipo de regla o decreto es concretar y decidir el tamaño además de la comunicación entre microservicios, en esencia conlleva a usar estadísticas con el objetivo de determinar elementos extensos que se encuentran dentro del mismo.

“Al usar microservicios se puede desplegar una aplicación grande como un conjunto de aplicaciones pequeñas que pueden ser desarrolladas, implementadas, ampliadas, manejadas y monitoreadas de manera independiente. La agilidad, la reducción de costos y la escalabilidad

granular, trae algunos desafíos y la complejidad de manejar sistemas distribuidos” (Vera-Rivera, 2018)

Otro atributo de calidad que compone un microservicio es la Cohesión, el mismo que hace referencia a la cantidad de tareas que realiza un microservicio y que se basa en una sola operación. A este atributo se le puede reconocer una ventaja, la cual es que no se lo puede medir con facilidad, pero el hecho de que sea difícil su medición no lo hace imposible, solo prolonga en cierta manera un poco más el proceso, la forma de medir la cohesión es creando y permitiendo la visualización de interfaces y comparando la posible similitud, comparando sus datos. Entonces queda claro que la cohesión caracteriza fuertemente a un microservicio.

“La arquitectura de microservicios presenta independencia, acoplamiento, cohesión, flexibilidad, despliegue y descubrimiento de servicios” (De la Cruz Vélez de Villa et al., 2019)

8.9.1 Patrones de Microservicios

“El enfoque de microservicios es un patrón de diseño de software que aprovecha los protocolos de redes para comunicar pequeñas aplicaciones que cumplen un objetivo específico.”(-chile & Arévalo del Río, 2016)

El objetivo de esta arquitectura es el poder solucionar ciertos problemas referentes a las Arquitecturas Orientadas a Servicios (SOA), por esto se elabora una interfaz gráfica de usuario fácil, construida básicamente a base de procesos unificados, y la metodología REST, además de esto se propusieron técnicas que posteriormente hicieron referencia a un enfoque monolítico, esto permitió el uso de tecnologías adecuadas y con una evolución mucho más evidente. Con esto se establecieron ventajas y desventajas acerca de este proceso, refiriéndose básicamente a la relación que tiene con la calidad, finalmente se pudo verificar que este enfoque o arquitectura da la facilidad de realizar una variación con respecto a la forma en que se pueda tener absoluto control de los procesos llevados en una aplicación, lo que da la ventaja de menorar considerablemente el porcentaje de errores y problemas.

Cuando se realizan proyectos de desarrollo de software a lo largo del tiempo tienen inconvenientes, en el momento de que el proyecto o la aplicación comienza a crecer y tomar forma, lo que produce que exista un exceso en cuanto al tamaño. Esto genera que todo el código existente mediante el desarrollo se desorganice y provoque una aglomeración de información, y producto de esto se genere un gran problema. Cuando el avance que tiene el proyecto crece sin ningún tipo de control produce en todos los componentes que lo constituyen lo que se

denomina acoplamiento o acumulación de datos y procesos, se desarrolló una alternativa para poder solventar y manejar el aumento que tienen estos proyectos, el cual es ocupar patrones de diseño de software los cuales contribuyan en la opción de facilitar el poder volver a estructurar la aplicación o el proyecto en el que se está trabajando, provocando menor acoplamiento en todo el código generado mediante el desarrollo, y finalmente así hacer crecer la fiabilidad en cuestión del bienestar del sistema.

“Este patrón de diseño se presenta como una alternativa moderna que brinda flexibilidad, escalabilidad e independencia a momento de diseñar y construir software.” (Quevedo-Avila et al., 2020)

Se puede identificar algunos de los patrones más importantes implementados de la arquitectura de microservicios:

- **Patrones de descomposición:** con este patrón se puede desintegrar una aplicación en busca de posibles fallos existentes.
- **Patrones de evaluación:** a este tipo de patrón se lo puede utilizar para realizar una prueba de rendimiento a la aplicación, en caso de necesitarla.
- **Patrones de interfaz de usuario:** este patrón se presenta como solución para colocar o rediseñar una nueva interfaz, según sea el requerimiento por parte del usuario o del propio sistema.
- **Patrones de seguridad:** este patrón interviene como un agente de seguridad para velar por la integridad de las peticiones que llegan al sistema o aplicación.
- **Patrones de despliegue:** este patrón verifica la viabilidad de un microservicio en cuanto a la comunicación que tiene con el servidor antes de registrarse en el mismo.

8.9.2 Arquitectura Orientada a Microservicios

“Es un nuevo patrón de arquitectura para el desarrollo de software, el cual permite mantener un conjunto de microservicios independientes. Cada uno de estos microservicios tienen como objetivo mantener la funcionalidad de un software de forma modular, cada microservicio funciona de forma independiente, apuntando a un área específica del modelo de datos.” (Muñoz Hernández, 2018)

Principalmente a donde apunta la arquitectura orientada a servicios, es precisamente a servicios de partición refiriéndose a los alcances del negocio desarrollado, SOA es estructurado

básicamente de funcionalidades creadas para fines empresariales, MSA igualmente tiene el objetivo de enfocarse en el tamaño del servicio.

8.9.3 Metodología de desarrollo de Microservicios

Los microservicios en la última década han sobresalido y han adquirido popularidad, por motivo de que grandes empresas se han visto en la necesidad de proporcionar servicios para sus usuarios y al percibir problemas que no se pueden solucionar fácilmente, sino es con el cambio de arquitectura tradicional hacia la arquitectura de microservicios, teniendo que migrar obligatoriamente sus sistemas e información, por otro lado se debe recordar también a las metodologías ágiles las cuales son fundamentales en el desarrollo de software ya que gracias a su implementación facilitan el resultado de un producto de buena calidad y confiable, entonces es necesario mencionar metodologías diseñadas exclusivamente para el desarrollo de microservicios, a continuación se presenta una metodología exclusiva para su desarrollo, obtenida a través de la búsqueda respectiva en fuentes bibliográficas.

8.9.3.1 Metodología MicroIoT

“Es una metodología para la definición, creación y despliegue de microservicios basada en metodologías ágiles, para soluciones de IoT desplegadas en AAL, la cual; alineada con el enfoque organizacional DevOps, el cual abarca adecuadamente el desarrollo y las operaciones sobre microservicios.”(Cabrera Alvarado & Cárdenas Cárdenas, 2018)

Con la cita mencionada anteriormente se evidencia que esta es una metodología diseñada con el fin de ayudar en la creación y desarrollo de microservicios, construida básicamente lineamientos de una metodología ágil, que va de la mano con otra metodología ágil como es DevOps que tiene un enfoque directo con la arquitectura de microservicios, además de esto también utilizan el modelo iterativo – incremental como herramienta para describir iteraciones con las que permiten que la metodología propuesta sea mucho más ordenada.

“En las diferentes actividades planteadas en MicroIoT, la metodología propuesta, intervienen diferentes actores o roles que deben cumplir las personas que desarrollen dichas actividades. Los principales roles que intervendrán durante el ciclo de vida planteado para cada entrega de software, en MicroIoT, pueden ser de dos tipos, roles internos como el caso de los roles de la organización de desarrollo, o externos como lo son los interesados, conocidos también como Stakeholders.”(Cabrera Alvarado & Cárdenas Cárdenas, 2018)

Como en otras metodologías, la creación y diseño de esta metodología se establecieron roles en cuanto se refiere a la organización que tiene que ver con su desarrollo, los cuales se presentan a continuación:

- Arquitecto de software
- Analista de requerimientos
- Equipo de desarrollo: en este equipo intervienen varios actores fundamentales que participan como en todo ámbito refiriéndonos en el desarrollo de software, un desarrollador, un tester y un administrador principal que se encargue de las bases de datos.

“MicroIoT se centra en profundizar la etapa de planificación de DevOps, desglosando la planificación en las actividades: i) análisis de requerimientos, ii) diseño de la entrega, iii) adecuación de la arquitectura y iv) validación del diseño y arquitectura de la entrega o solución.”(Cabrera Alvarado & Cárdenas Cárdenas, 2018)

8.9.3.2 Metodología DevOps

“DevOps desde su origen se ha entendido no únicamente como un conjunto de herramientas o procesos, sino como una cultura organizativa. Dentro de esta nueva dinámica de trabajo, se trata poner en alza algunos aspectos del trabajo diario mejora en la documentación de procesos, buena intercomunicación y colaboración entre equipos, reparto de responsabilidades, aprender de errores pasados, estar abiertos a nuevas ideas que, por dejadez o mala gestión de los tiempos, han pasado a un segundo plano provocando que los equipos no sean tan eficientes.”(Marco, 2016)

El objetivo principal de DevOps es optimizar a toda costa la agilidad de la comunicación entre servicios, potencia las interrelaciones los equipos de desarrollo establecidos y cada una de sus tareas, con esto se consigue evadir problemas que tengan relación con el proyecto a realizar y que pueda afectar la calidad del software y la entrega posterior del producto final.

DevOps al ser parte de un grupo de herramientas o procesos ágiles, representa una estructura organizacional mediante la cual mejora la comunicación y relación entre equipos, con el objetivo de superar de manera rápida errores ya cometidos anteriormente.

Tabla 4. Tabla comparativa – MicroIoT y DevOps

<p>Metodología MicroIoT</p>	<ul style="list-style-type: none"> • Es una metodología alineada con DevOps y diseñada concretamente para la creación y desarrollo de microservicios que se centra específicamente en una etapa de DevOps la cual es la planificación, dividiéndola en varias actividades como son la determinación de requerimientos, verificación de la entrega, diseño de la arquitectura y validación. • Esta metodología se la puede adaptar a cualquier proceso de desarrollo de software de manera sencilla según las necesidades surgidas. • MicroIoT cuenta con etapas sencillas para desarrollar software específicamente para microservicios como son la identificación, definición, verificación y operación de los mismos. • Permite adaptar los roles del equipo de desarrollo según el propósito de software.
<p>Metodología DevOps</p>	<ul style="list-style-type: none"> • DevOps tiene varias fases como planificación, desarrollo, entrega y funcionamiento, y para cada una de estas etapas se necesita varios desarrolladores que estén pendientes y velando por el correcto desempeño de la aplicación, además de solucionar errores oportunamente, es decir se necesita un equipo de desarrollo completo. • Los procesos realizados a través de esta metodología necesitan supervisión continua por parte del equipo de desarrollo., es por esto que se dividen roles para cubrir cada una de estas actividades. • Necesita obligatoriamente tener un cliente que determine requerimientos, los cuales posteriormente sean cumplidos por el equipo de desarrollo.

Fuente: Christian Guevara (Investigador)

Mediante la revisión bibliográfica se han determinado dos metodologías para el desarrollo de microservicios como son MicroIoT y DevOps, mediante las cuales se ha seleccionado MicroIoT para el desarrollo de este prototipo por todas las ventajas que ofrece con respecto a la capacidad de adaptarse a cualquier propósito de desarrollo de software, permitiendo disminuir los roles del equipo de desarrollo y al estar relacionado con DevOps con respecto a enfocarse en una sola etapa como es la planificación, facilita la tarea de desarrollar un sistema como demostración del trabajo real que realizan los microservicios en una aplicación.

8.10 Términos Básicos

- ✓ **Aplicación:** Es un programa desarrollado con el propósito de funcionar como herramienta para realizar tareas determinadas.
- ✓ **Framework:** Es una estructura que nos facilita un espacio o entorno para distar código en un lenguaje determinado, esto produce ahorro de tiempo y recursos a todo desarrollador que lo utilice.
- ✓ **Base de datos:** Es un sistema donde se almacena información de manera organizada y estructurada.
- ✓ **Servidor:** Es un entorno estructurado para procesar peticiones y entregar solicitudes a otras computadoras que se les puede nombrar usuarios.
- ✓ **Prototipo:** Es un ejemplo de una primera versión o prueba con el objetivo de ser demostrado mediante su implementación.
- ✓ **Software:** El software nos facilita dirigir y encaminar recursos que requiere un SO de un ordenador para controlar programas, además de esto sirve como vía de interacción y comunicación entre el usuario y el hardware.

9. PREGUNTAS CIENTÍFICAS O HIPÓTESIS

La selección de metodologías planteadas como propuesta permitirán innovar el desarrollo de soluciones informáticas orientadas a microservicios en la Universidad Técnica de Cotopaxi.

9.1 Verificación de la propuesta

Para la verificación de la propuesta se utilizó Juicio de Expertos como técnica para comprobar la confiabilidad del proyecto, además de esto se consideró un cuestionario de validación como instrumento de recolección de datos, cuyos resultados fueron aplicados con el método de Alfa de Cronbach para su correspondiente fiabilidad.

9.2 Juicio de expertos

“El juicio de expertos se define como una opinión informada de personas con trayectoria en el tema, que son reconocidas por otros como expertos cualificados en éste, y que pueden dar información, evidencia, juicios y valoraciones.”(Jazmine Escobar-Pérez, 2008)

El juicio de expertos es un método de validación, el cual sirve para comprobar la fiabilidad de una investigación en concreto, para esta verificación se deben seguir ciertos pasos como recolectar diversas opiniones de personas que tengan un conocimiento considerable, aceptable y confiable sobre el tema, en nuestro caso se optó por recurrir al conocimiento de 3 expertos en

el tema del desarrollo de software y arquitectura de microservicios, con el fin de garantizar la validez y fiabilidad de esta investigación.

9.3 Alfa de Cronbach

Básicamente Alfa de Cronbach es una herramienta con la que se facilita la medición de confiabilidad en relación a la información recolectada mediante un instrumento de evaluación, además de esto como complemento, nos brinda la posibilidad de una medición también en cuanto a la magnitud o escala de los ítems que forman parte de la estructura de dicho instrumento.

10. METODOLOGÍAS Y DISEÑO EXPERIMENTAL

10.1. Tipos de Investigación

10.1.1 Investigación bibliográfica

Este tipo de investigación se la trata como técnica que se utiliza para recolectar información previamente seleccionada que se realiza de manera ordenada por medio del análisis y revisión de la lectura de revistas, libros y fuentes bibliográficas confiables que brindan información verídica, este último punto es demasiado importante ya que este tipo de investigación depende al cien por ciento de la información que se obtiene.

Aprovechando la ventaja que conlleva utilizar este tipo de investigación se podido investigar temas necesarios para la realización de esta investigación, como gestores de bases de datos, servidores, lenguajes de programación y frameworks, todas estas partes mencionadas son fundamentales para el desarrollo de la idea del prototipo establecido previamente, además de esto se investigó metodologías que sirven como punto de partida para desarrollar microservicios y para adentrarse a conocer más sobre esta arquitectura.

10.2 Metodología aplicada al proyecto

10.2.1 ¿Qué es una metodología de desarrollo?

Las metodologías ágiles son conocidas por su capacidad de adaptación en la manera de trabajar, independientemente de los requerimientos del proyecto en cuestión, logrando así mayor rapidez y flexibilidad al momento de adaptarse al proyecto y su constante desarrollo.

Prácticamente este tipo de metodologías en el ámbito del desarrollo de software proporciona una estructura organizada, gracias a lo cual un equipo de trabajo o desarrollo puede tomarlo como punto de partida en el momento de construir aplicaciones de la mejor manera posible, con esto se asegura que el producto final entregado sea de la completa aceptación y agrado por parte

del cliente o usuario, es indispensable desarrollar cualquier tipo de aplicación a partir de una metodología para tener control y brindar seguimiento a cada avance que se tenga en el proyecto a realizar.

Con el pasar de los años se han presentado varias metodologías de software, las cuales satisfacen un determinado campo:

- Modelo Iterativo - Incremental
- Metodología en Cascada
- Metodologías Ágiles
- Metodología de Prototipado
- Metodología en Espiral

10.2.2 Metodologías ágiles

El utilizar metodologías ágiles ofrece ventajas muy importantes y mejores resultados en donde se las aplique ya que para empezar mejora notablemente la calidad del producto final, ya que estas metodologías incentivan la actividad entre el equipo de desarrollo en la proceso de buscar los mejores resultados para dicho producto, con esto se logra mayor satisfacción por parte del cliente ya que se lo involucra en el proceso de desarrollo con demostraciones del progreso del producto, lo cual genera confianza, y por último se reduce los costos ya que brinda garantías a lo largo del proceso y desecha cualquier posibilidad de fracaso en el proyecto.

Este tipo de metodologías permiten identificar y definir de mejor manera las labores que se están realizando o se vaya a realizar a futuro en el proyecto para poder ser implementadas según los requerimientos que se establezcan, obteniendo resultados favorables en el proyecto que se esté trabajando como flexibilidad y eficiencia, estas características provocan que se obtenga mayor calidad y el costo establecido a un principio se reduzca considerablemente.

En este proyecto ha sido necesario utilizar practicas agiles como el modelo iterativo – incremental para determinar funcionalidades, las cuales componen el prototipo diseñado, también se utilizó la metodología Scrum, misma que permitió establecer los roles correspondientes para el equipo de desarrollo, además de esto se usaron varias prácticas ágiles como son el Product Backlog para poder detallar los requisitos necesarios para la implementación del proyecto, historias de usuario con el objetivo de especificar los requisitos establecidos previamente, y por último se utilizaron Sprints para poder detallar el incremento que tuvo el proyecto mediante iteraciones.

10.2.3 Modelo iterativo incremental

El objetivo de este modelo es el poder disminuir el riesgo que se genera en los requerimientos del producto y el usuario por inconvenientes presentados o encontrados oportunamente en la etapa de recolección de requerimientos, se puede decir que este modelo es nacido o derivado del modelo en cascada.

Ventajas de la utilización del modelo Iterativo – Incremental:

- Brinda ayuda para establecer aún efectivos son los procesos, además de que tan bueno es el producto final.
- Facilita el estudio para posteriormente avanzar y estructurar los procesos correspondientes para un determinado entorno.
- Se le puede dar mantenimiento al producto a fin de evitar posibles problemas que puedan afectar en un futuro a dicho producto.

10.2.4 Metodología devops

“DevOps nace como un nuevo paradigma en contraposición al modelo de desarrollo de software tradicional con el objetivo de mejorar la comunicación y la colaboración entre los diferentes equipos de trabajo y aumentar así su eficiencia.”(RIVERA CARO, 2019)

Esta metodología tiene ciertos principios, los cuales la identifican y la hacen única a comparación de las demás, dentro de sus características más importantes se reconoce a los clientes de un determinado proyecto y el nivel de satisfacción que este pueda tener, ya que es algo fundamental y se encuentra al finalizar cada etapa del proceso en cuestión de desarrollo y operación de DevOps, la responsabilidad es algo fundamental en cualquier aspecto y más para esta metodología ya se debe tener en cuenta que al momento de elegir DevOps se debe proponer una garantía de éxito a lo largo de todos los procesos propuestos por la metodología, otro principio que se debe tomar en cuenta es el hecho de que el proceso debe estar en constante control con respecto a la mejoría, es decir, es necesario establecer un control para que los procesos empresariales estén en su punto máximo de rendimiento, y finalmente una ventaja que resalta en este proceso fundamental es que automatiza por completo el sistema, ya que su objetivo principal es que los procesos u operaciones que se realizan en el sistema sean mejorados y automatizados en cada etapa que forma parte de la metodología.

DEVOPS Y SU CICLO DE VIDA

DevOps extiende este acercamiento ágil a todo el ciclo de vida de entrega de software y también a todos los interesados, incluyendo las líneas de negocio y los equipos de operaciones, en lugar de solo dejarlo para los equipos de desarrollo y QA.(RETAMAL VALENZUELA, 2019)

La metodología DevOps es estructurada mediante las siguientes fases:

- Planificación
- Codificación
- Construcción
- Pruebas
- Entregas
- Despliegue
- Monitoreo

Según Retamal (2019) el propósito original del movimiento DevOps fue remover las barreras entre los equipos de desarrollo y operaciones. La falta de comunicación y confianza entre estos equipos desafió su habilidad de desplegar software. La falta de comunicación efectiva resultante del trabajo en silos, hicieron difícil para los equipos de desarrollo y los interesados de negocio recibir la retroalimentación de los usuarios. Sin un mecanismo efectivo de retroalimentación de los usuarios, estos equipos están incapacitados para responder y beneficiarse de esta retroalimentación.

Aunque el nombre “DevOps” sugiere que son capacidades orientadas a desarrollo y operaciones, este proceso es una capacidad empresarial que incluye a todos los interesados en una organización incluyendo los dueños del negocio, arquitectura, diseño, desarrollo, QA, operaciones, seguridad, partners y proveedores.(RETAMAL VALENZUELA, 2019)

10.2.5 Metodología MicroIoT

“MicroIoT representa una metodología que permite la creación de aplicaciones basadas en microservicios que a más de ser aplicada a los dominios de IoT y AAL, puede adaptarse a otros de manera muy sencilla. En este trabajo se ha propuesto el uso de microservicios, debido a las características propias de los mismos (resiliencia, disponibilidad, tolerancia a fallos, fácil escalabilidad), lo que concuerda plenamente con las necesidades de IoT para AAL”(Cabrera Alvarado & Cárdenas Cárdenas, 2018)

Se seleccionó MicroIoT con el propósito de establecer requerimientos y funcionalidades del prototipo de microservicios que se propuso realizar al principio de la investigación, esta metodología fue elegida como alternativa para el desarrollo, ya que contiene el enfoque organizacional de DevOps que es la metodología adecuada para lo que es el desarrollo de microservicios, además de la fácil adaptación a diferentes procesos, con la unificación de estas metodologías se logró obtener ventajas y beneficios importantes como s fácil escalabilidad y tolerancia a fallos.

Esta metodología se adaptó para el desarrollo de un sistema con varias funcionalidades principales, con la investigación bibliográfica realizada se pudo determinar que esta metodología era la apropiada para desarrollar microservicios por motivo de que está diseñada y enfocada precisamente para eso, en consecuencia de esto tomando el modelo organizacional y ciclo de vida de Devops ya que es parte de su estructura, y que tiene como producto se obtiene que en cada iteración o etapa se debe entregar una nueva función a las tareas planteadas e importancia a cada avance del software.

10.2.5.1 Ciclo de vida

Este ciclo de vida de esta metodología está compuesto de siete etapas o fases, pero para el desarrollo de nuestro prototipo se ha decidido adaptar estas etapas a los requerimientos planteados, quedándonos con estas fases finales:

- **Análisis de requerimientos**

En esta actividad se determinó todos los requerimientos y funcionalidades principales pertenecientes a microservicios para el desarrollo del prototipo.

- **Arquitectura del sistema**

El propósito de esta actividad es acomodar la arquitectura de los microservicios y su respectiva relación.

- **Desarrollo y pruebas**

En esta actividad se realiza la respectiva codificación y desarrollo de la aplicación o sistema, con un control eventual por parte del equipo de desarrollo.

- **Operaciones**

En esta actividad se realiza la verificación y control del desempeño de la aplicación, con el objetivo de que se cumplan todos los requerimientos establecidos.

Todas estas actividades son parte de esta metodología y se consideraron fundamentales para el que se lleve a cabo el desarrollo y mantenimiento del prototipo.

10.2.5.2 Identificación de microservicios

Los microservicios tuvieron que ser especificados luego de pasar por las etapas y actividades anteriores, estos se determinan tomando en cuenta factores como el tamaño del equipo, escalabilidad y el tipo de datos que se propongo enviar o recibir.

Aquí se tiene la posibilidad de fraccionar a un microservicio en servicios más pequeños, y de igual manera a un servicio pequeño volverlo más grande según su funcionalidad, todo esto obviamente se puede realizar verificando la necesidad.

La identificación de los microservicios en esta instancia es importante para la continuación del proyecto ya que es un punto fundamental para seguir con los demás procesos restantes.

10.2.5.3 Definición de microservicios

Para definir un microservicio se determinó varios aspectos importantes que no se pueden olvidar, como son servicios funcionales, los cuales son números y letras que pueden utilizarse en una aplicación, esto sirve como herramienta para tomar medida de identificación, por ejemplo, ids como identificadores únicos de desarrollo, y servicios no funcionales los cuales se refieren a servicios que no tienen tareas que representan algún tipo de operación, se puede mencionar a un sencillo login como ejemplo del proceso de operación.

Mencionando esto en los microservicios se establecieron componentes que son fundamentales al momento de implementarse en algún sistema o aplicación, en este caso nuestro prototipo, los componentes necesarios para el desarrollo del prototipo se definieron desde la parte del backend, eligiendo las bases de datos adecuadas para cada microservicio y a su vez el servidor donde todos los servicios se iban a registrar previamente a su levantamiento, una vez ya se tuvo establecido el backend se procedió a determinar el frontend, tomando en cuenta un lenguaje de programación que sea compatible con la estructura y desarrollo de los microservicios y un framework que tenga relación con el lenguaje para su desarrollo.

10.2.5.4 Verificación de microservicios

En esta etapa se verificó el diseño de la estructura una vez ya se hayan establecido los microservicios a implementarse, cada microservicio se controló que cumpla con ciertas normas

como tener determinada una sola tarea, y que este conformado por backend y frontend, además de esto es indispensable que estén trabajando de manera única y que haya una correcta comunicación que influya en el trabajo de manera positiva, además del desarrollo y comunicación con otros servicios.

10.2.5.5 Operaciones

En esta etapa se establecieron dos etapas que tienen relación con la fase de operación de la metodología DevOps, el control del desempeño de las aplicaciones que han sido puestas en producción y tener en cuenta los requerimientos establecidos previamente por el cliente o usuario, la ventaja de esto es que se puede reconocer cualquier tipo de problema en el prototipo o sistema, con esto reaccionar de manera ágil y eficaz con el objetivo de mejorar el rendimiento y evitar inconvenientes referentes a la elevación de costos.

10.2.5.6 Características en el desarrollo de microservicios

Una vez establecidos los microservicios a utilizar en el prototipo y establecer una estructura principal basada en la metodología propuesta, se consideró la composición básica de los microservicios los cuales tienen cualidades importantes que los caracteriza, mismos que se detallan a continuación.

Granularidad

Esta etapa es indispensable cuando se trata de diseñar la arquitectura de microservicios en cuanto al tamaño de cada uno, por esta fase se debe empezar en primer lugar antes de ejecutar cualquier otro proceso, ya que con una correcta granularidad es posible aprovechar al cien por ciento las ventajas que ofrece esta arquitectura, ya que cada microservicio se encarga de una sola función con esta etapa se controla el avance en cuestión de desempleo de cada servicio.

Modularidad

Con esta característica cuenta todo microservicio bien desarrollado, ya que, por ser servicios autónomos, tienen la facilidad de constituir un desarrollo independiente, además de esto la ventaja más importante y que más sobresale es en cuestión de fallo o errores ya que si por algún motivo un microservicio tiene problemas de desempeño, esta acción no debería tener la capacidad de intervenir en los procesos de los demás microservicios, y debe continuar trabajando normalmente.

Escalabilidad

Al ser una aplicación modular se puede escalar con facilidad, es decir interrelacionarse entre servicios según la necesidad, ya que al contener a su cargo una única y distinta tarea también tiene la facilidad de integrarse y comunicarse con otros al realizar sus respectivas peticiones, para así poder demostrar un trabajo independiente, pero a la vez una exitosa comunicación.

Rapidez de actuación

Una de las ventajas más importantes que conlleva la utilización de esta arquitectura y que posee el prototipo de esta investigación, es el hecho de que gracias a que los microservicios son muy pequeños nos brinda la ventaja de poner en ejecución la aplicación o sistema lo más rápido posible, gracias a esto se tiene una considerable disminución de costos al momento de desarrollar y entregar el producto final.

Mantenimiento

Esta es otra característica con la que cuenta esta arquitectura y la última que compone este prototipo, ya que gracias al trabajo independiente y a la estructura que compone un microservicio, cuando trabajan en conjunto para lograr un objetivo mayor y al encontrarse con algún tipo de error, es fácil identificarlo ya que el desarrollador al conocer como está compuesta la aplicación reduce el tiempo de solución, es decir tiene tolerancia a fallos.

Finalmente, para la validación de la propuesta se ha realizado una encuesta, esta técnica se la utilizó en conjunto con juicio de expertos, mediante el cual se ha validado dicha propuesta con el método de Alfa de Cronbach.

11. ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS

11.1 Arquitectura del Prototipo

En esta sección se van a presentar los resultados del prototipo desarrollado a través de las herramientas seleccionadas y la aplicación de las practicas agiles descritas en la sección de la metodología.

11.1.1 Título del Prototipo

App - Microservicios

11.1.2 Qué hace el Prototipo

El presente prototipo consta de cuatro microservicios, los cuales están conectados a diferentes bases de datos con el propósito de demostrar su desempeño, cada microservicio consta con

opciones y tareas establecidas como el poder crear, editar, eliminar, entre otras actividades variantes.

11.1.3 Objetivos

Se establecieron 4 funcionalidades importantes:

- Microservicio Usuario (Alumnos)
- Microservicio Cursos
- Microservicio Exámenes
- Microservicio Respuestas

11.1.4 Fechas críticas

- Primera funcionalidad - Microservicio Usuarios (Alumnos) fue establecido del 6 de noviembre del 2020 al 27 de noviembre del 2020.
- Segunda funcionalidad - Microservicio Cursos fue establecido del 28 de noviembre del 2020 al 18 de diciembre del 2020.
- Tercera funcionalidad - Microservicio Exámenes fue establecido del 19 de diciembre del 2020 al 08 de enero del 2021.
- Cuarta funcionalidad - Microservicio Respuestas fue establecido del 09 de enero del 2021 al 29 de enero del 2021.

11.1.5 Miembros del equipo de desarrollo

Tabla 5. Rol del equipo – Scrum Master

Nombre	Ing. Verónica Consuelo Tapia Cerda
Rol	Jefe de Proyecto
Responsabilidad	Revisión de la aplicación
Información Contacto	veronica.tapia@utc.edu.ec

Tabla 6. Rol del equipo – Scrum Team

Nombre	Christian Paúl Guevara Naranjo
Rol	Desarrollador

Responsabilidad	Generar el código para los microservicios del prototipo.
Información Contacto	christian.guevara9@utc.edu.ec

11.1.6 Product backlog

Mediante reuniones con el jefe del proyecto se establecieron 4 Sprints los cuales se muestran en la siguiente tabla.

Tabla 7. Sprints a desarrollar

Sprint	Tarea	Tiempo	Fecha de entrega	Descripción
1	Microservicio Usuario (Alumnos)	3 semanas	27 de noviembre del 2020	Permite gestionar la información de los alumnos.
2	Microservicio Cursos	3 semanas	18 de diciembre del 2020	Permite gestionar la información de los cursos.
3	Microservicio Exámenes	3 semanas	8 de enero del 2021	Permite gestionar la información de los exámenes
4	Microservicio Respuestas	3 semanas	29 de enero del 2021	Permite gestionar la información de las respuestas.

11.1.7 Prácticas ágiles usadas durante el desarrollo del proyecto

Product backlog

Es un listado ordenado de todos los requerimientos que se necesita para poder desarrollar el prototipo, aquí se establecen las funcionalidades que se van a implementar.

Historias de usuario

Es una práctica ágil donde básicamente se presentan los requerimientos detallados de muy buena manera, se deben especificar las funcionalidades, roles y resultados esperados.

Roles

En el presente proyecto se consideró varios roles para el equipo de trabajo, donde se muestra el jefe de proyecto y el desarrollador.

Sprints

Con la utilización de esta práctica ágil con el cual se puede detallar y mostrar el incremento que va ha tenido el proyecto que en este caso son iteraciones.

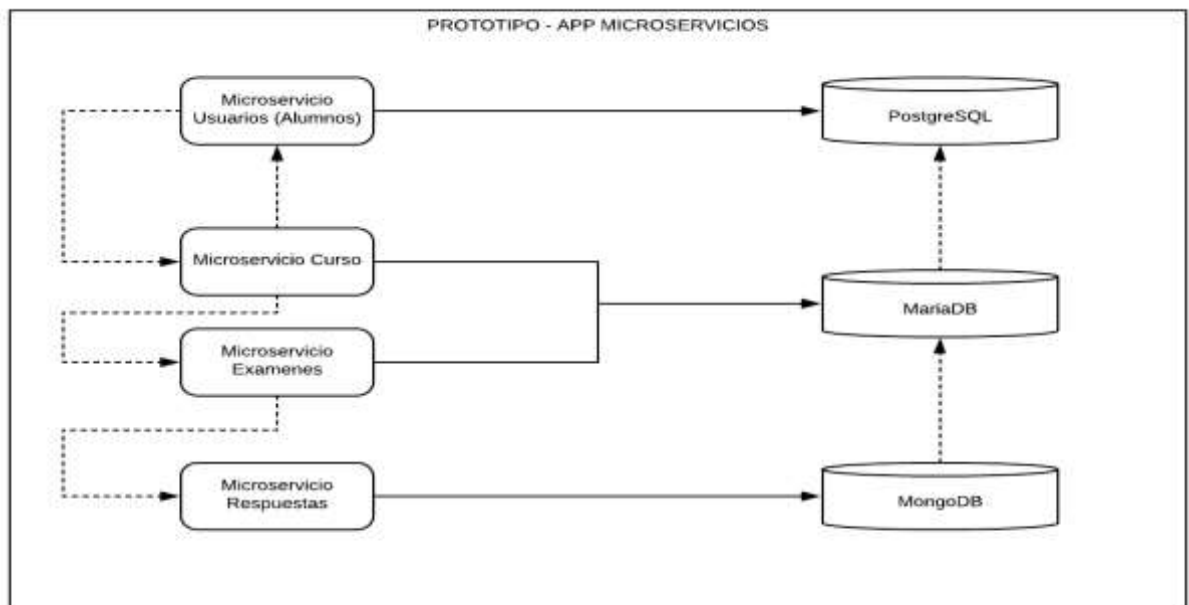
11.1.8 Tecnologías usadas durante el desarrollo del proyecto

En el desarrollo del prototipo se utilizaron varias tecnologías tanto para frontend y para backend, en el caso del frontend se utilizó el lenguaje de programación JavaScript y como framework Angular que es una tecnología que tiene una relación directa con el lenguaje mencionado anteriormente, en el caso del backend se trabajó en primer lugar con Spring Boot y eureka que es un anexo de esta tecnología, esto como servidor para el registro de cada microservicio, además de esto para complementar la funcionalidad del tema se utiliza varias bases de datos, tales como PostgreSQL, MongoDB y MariaDB.

11.1.9 Conexión de los Microservicios con sus respectivas bases de datos

En el presente diagrama se representa la conexión que se realizó entre los 4 microservicios establecidos y cada una de las bases de datos seleccionadas para su posterior programación y desarrollo.

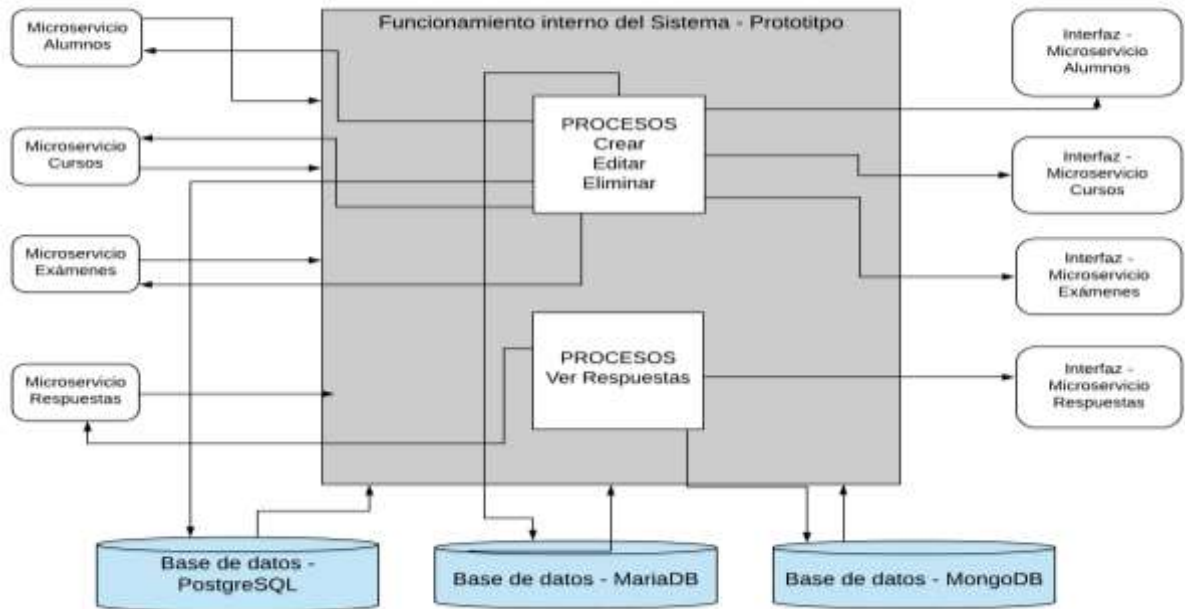
Figura 1. Diagrama - Conexión Microservicio - Base de Datos



11.1.10 Diagrama de bloques

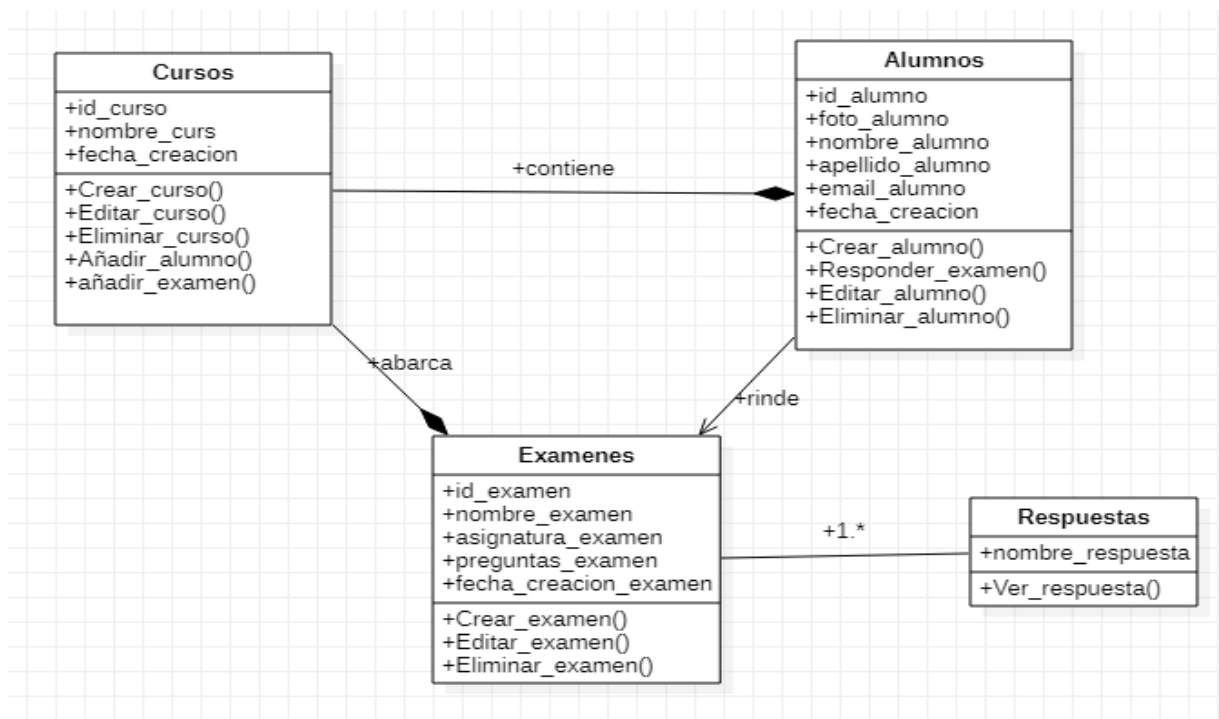
En el siguiente diagrama de bloques se representa el funcionamiento interno del prototipo de microservicios, con los 4 módulos que representan a las funcionalidades implementadas.

Figura 2. Diagrama de bloques – Módulos del prototipo



11.1.11 Modelo de datos

Figura 3. Modelo Relacional



11.1.12 Release Plan Final

En este apartado se describen todos los Sprints o funcionalidades concretadas que finalmente se van a entregar.

Tabla 8. Release Plan Final

Plan final			
Primer sprint – Estimación inicial	Microservicio Usuarios (Alumnos)	H. U	- Crear - Editar - Eliminar
Segundo sprint - Estimación inicial	Microservicio Cursos	H.U	- Crear - Editar - Eliminar
Tercer sprint - Estimación inicial	Microservicio Exámenes	H.U	- Crear - Editar - Eliminar
Cuarto sprint – Estimación inicial	Microservicio Respuesta	H. U	- Ver Respuesta

11.1.13 Detalles de todas la Historias de usuario

Tabla 9. Historia de usuario 1 – Crear Usuario (Alumno)

HISTORIA DE USUARIO(HU)			
Código HU:	HU0001	Fecha:	0/0/0
Sprint:	1	Prioridad:	Alta
Actores:	Administrador	Puntos:	3
Descripción: Como administrador deseo poder “Crear” y registrar un alumno nuevo al sistema.			
Detalles de la HU:			
<ul style="list-style-type: none"> • El administrador selecciona la opción crear de la sección “Alumnos”. • El administrador llena todos los campos con la información del nuevo alumno. • Por medio de código se verifica que todos los campos estén llenos para crear el alumno. • La información se registra con éxito en la base de datos correspondiente. 			

Restricciones: Solo el administrador del prototipo puede crear un nuevo alumno.

Criterios de aceptación:

- Los campos deben ser completados en su totalidad.

DoD:

- Todos los campos presentes en el formulario deben tener sus respectivas validaciones.
- La información digitada debe estar guardada y registrada en las bases de datos que corresponda.
- Debe existir un mensaje el cual notifique que los datos se registraron con éxito.

Tabla 10. Historia de usuario 2 – Editar Usuario (Alumno)

HISTORIA DE USUARIO(HU)			
Código HU:	HU0002	Fecha:	0/0/0
Sprint:	1	Prioridad:	Alta
Actores:	Administrador	Puntos:	3
Descripción: Como administrador deseo poder editar un alumno previamente registrado en el sistema.			
Detalles de la HU:			
<ul style="list-style-type: none">• El administrador selecciona la opción “Editar” de la sección “Alumnos”.• El administrador modifica la información deseada en el formulario del alumno.• Por medio de código se verifica que todos los campos estén llenos para editar la información del alumno.• La información del alumno se modifica con éxito y se registra en la base de datos.			
Restricciones: Solo el administrador del prototipo puede editar la información de un alumno.			
Criterios de aceptación:			
<ul style="list-style-type: none">• Los campos del formulario deben completarse en su totalidad para que los cambios realizados se guarden.			
DoD:			
<ul style="list-style-type: none">• Todos los campos presentes en el formulario deben tener sus respectivas validaciones.			

- La información modificada del alumno debe registrarse en la base de datos correspondiente.
- Debe existir un mensaje el cual notifique que la información del alumno se registró con éxito.

Tabla 11. Historia de usuario 3 – Eliminar Usuario (Alumno)

HISTORIA DE USUARIO(HU)			
Código HU:	HU0003	Fecha:	0/0/0
Sprint:	1	Prioridad:	Alta
Actores:	Administrador	Puntos:	3
Descripción: Como administrador deseo poder eliminar un alumno previamente registrado en el sistema.			
Detalles de la HU:			
<ul style="list-style-type: none"> • El administrador selecciona la opción “Eliminar” de la sección “Alumnos”. • El administrador elimina del sistema al alumno seleccionado. • La información del alumno eliminado se borra de la base de datos. 			
Restricciones: Solo el administrador del prototipo puede eliminar un alumno del sistema.			
Criterios de aceptación:			
<ul style="list-style-type: none"> • Previamente debe existir alumnos registrados en el sistema para poder eliminar. 			
DoD:			
<ul style="list-style-type: none"> • Debe existir un alumno para poder eliminarlo del sistema. • La información del alumno eliminado del sistema debe borrarse por completo en la base de datos. • Debe existir un mensaje el cual notifique que el alumno se eliminó con éxito del sistema. 			

11.1.14 Pruebas realizadas

Resultados del primer sprint “Microservicio Usuario (Alumnos)”

A continuación, se presenta los resultados obtenidos por parte de la primera funcionalidad el cual corresponde al primer sprint de Microservicio Alumnos, además de esto se presenta las tareas que realiza el Sprint como crear, editar y eliminar un alumno de sistema.

Resultados del análisis en el sprint “Microservicio Usuario (Alumnos)”

Resultados de la implementación en el sprint “Microservicio Usuario (Alumnos)”

Implementación de la historia de usuario “Crear Alumnos”.

En este apartado se presenta los resultados obtenidos mediante la implementación de la funcionalidad y diseño de la sección “Alumnos”, donde si se procede a presionar el botón “Crear”, el prototipo presenta la interfaz con un formulario donde se puede visualizar todos los campos, en los cuales se deben llenar con la información del alumno posteriormente a registrar o crear, además de eso en la interfaz también se presenta el segundo botón “Crear” mediante el cual si se presiona se procederá a guardar toda la información registrada en la base de datos correspondiente.

Figura 4. Implementación de “Crear Alumno”



Implementación de la historia de usuario “Editar Alumnos”.

En este apartado se presenta los resultados obtenidos mediante la implementación de la funcionalidad y diseño de la sección “Alumnos”, en el cual si se selecciona algún alumno registrado previamente en algún curso del sistema y se presiona el botón “Editar” la interfaz le va a presentar un formulario donde constan todos los datos e información del alumno que se seleccionó para modificar, para luego una vez realizadas todas las modificaciones pertinentes se procede a presionar el botón “Editar”, finalizando así el proceso de edición y guardado.

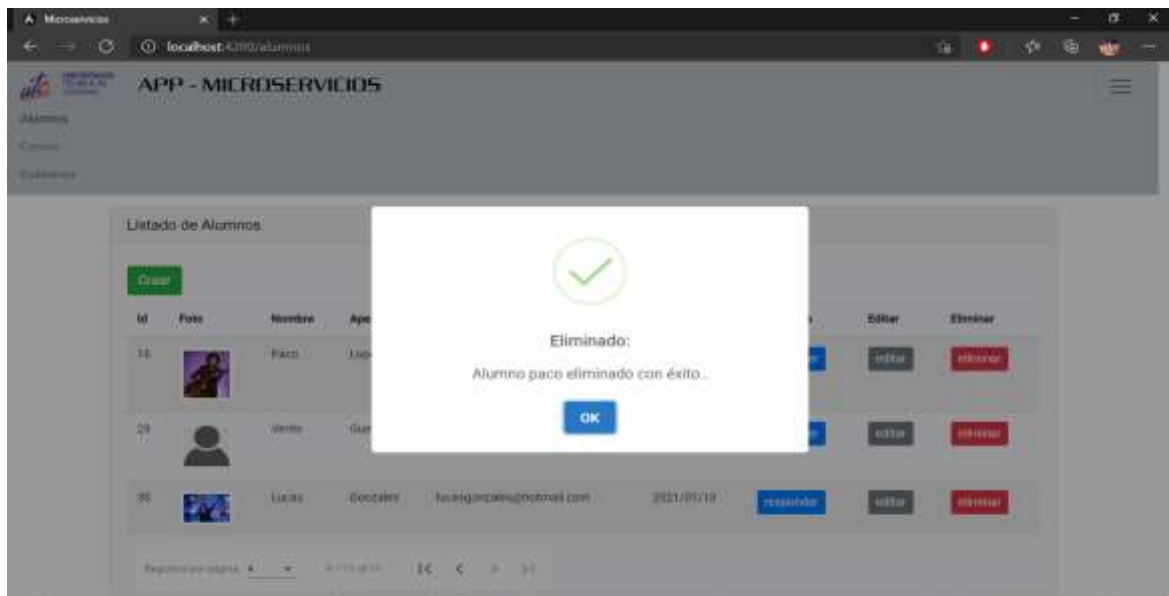
Figura 5. Implementación de “Editar Alumno”



Implementación de la historia de usuario “Eliminar Alumnos”.

En este apartado se presenta los resultados obtenidos mediante la implementación de la funcionalidad y diseño de la sección “Alumnos”, donde la interfaz presenta el botón “Eliminar”, mediante el cual si se presiona se procede a eliminar el alumno seleccionado, la interfaz le presenta un mensaje de alerta donde le indica si está seguro de querer eliminar al alumno, se confirma la petición y finalmente dicho alumno es eliminado de la interfaz y de la base de datos donde se alojaba toda su información.

Figura 6. Implementación de “Eliminar Alumno”



Validación de la propuesta

En este apartado se realizaron pruebas de funcionalidad en las que se aplicó casos de pruebas, a continuación, se describen las pruebas realizadas.

Pruebas realizadas en el sprint “Microservicio Usuarios (Alumnos)”

En esta sección se presentan las pruebas realizadas del primer Sprint.

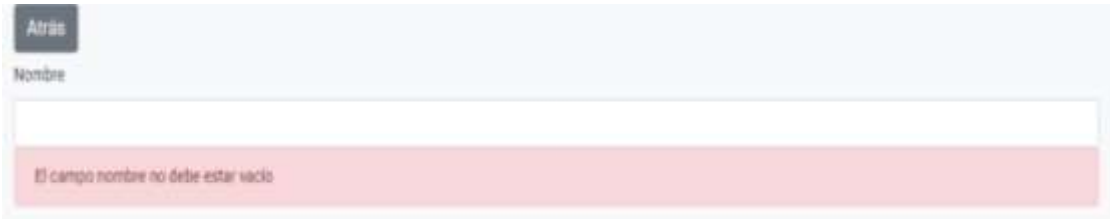

Objetivo

Diseñar el plan de pruebas de la historia de usuario “Crear Alumno”, “Editar Alumno” y “Eliminar Alumno”, todo esto se realiza con el objetivo de comprobar mediante el arrojo de resultados y la evaluación respectiva, si tendrá la posibilidad de ser o no superada.

Alcance

Mediante el plan de pruebas implementado se tiene la posibilidad de determinar y evaluar diferentes etapas del prototipo que se desarrolló, todo esto enfocado a las historias anteriormente mencionadas en el objetivo.

Tabla 12. Caso de Prueba CP_001

# Caso de Prueba		CP_001	
# Historia de Usuario	HU01	Fecha	10/02/2021
Descripción	Tiene por objetivo la respectiva verificación hacia el administrador del prototipo, es decir si tiene la posibilidad o no de poder crear nuevos alumnos y registrar esa información en la base de datos.		
Condiciones de Ejecución	El administrador presiona la opción “Crear” de la sección alumnos.		
Entradas	<ol style="list-style-type: none"> 1. Digita todos los datos del alumno 2. Presiona el botón “Crear” 		
Resultados Esperados 1	Si en el formulario donde están los datos del alumno a registrar existe uno sin completar, el administrador podrá visualizar mensajes emergentes donde falten campos por llenar.		
Resultado reflejado 1			
			
Resultados Esperados 2	Si el administrador en el campo “Foto” ingreso un archivo que sea diferente a formato imagen, se le presentara un mensaje emergente “Error al cargar la foto, el archivo debe ser de tipo imagen”		
Resultado reflejado 2			
			


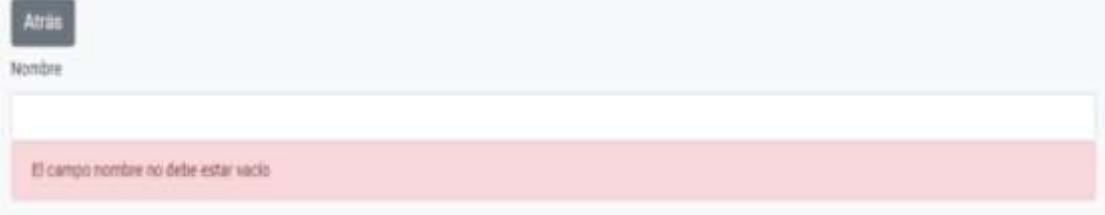


Resultados Esperados 3	Si el administrador cumplió con todos los requisitos para crear un nuevo alumno, como llenar todos los campos y en el campo “foto” subir un archivo de tipo imagen, al presionar el botón “foto” subir un archivo de tipo imagen, al presionar el botón crear se le presentará un mensaje “Alumno creado con éxito”.
Resultado reflejado 3	
Resultados Esperados 4	Si el administrador decide no crear un alumno, una vez estando en el formulario presiona el botón “Atrás” y regresa a la pantalla principal de la sección alumnos.
Evaluación de la Prueba	SUPERADA
Responsable	Tester del Equipo de Trabajo (Christian Guevara)


Tabla 13. Caso de Prueba CP_002

# Caso de Prueba	CP_002		
# Historia de Usuario	HU02	Fecha	10/02/2021
Descripción	Tiene por objetivo la respectiva verificación hacia el administrador del prototipo, es decir si puede o no editar a información de cualquier alumno registrado en algún curso del sistema.		
Condiciones de Ejecución	El administrador presiona la opción “Editar” de la sección alumnos.		
Entradas	<ul style="list-style-type: none"> - Modifica la información deseada del alumno. - Presiona el botón “Editar” 		

<p>Resultados Esperados 1</p>	<p>Si en el formulario donde están los datos del alumno a registrar existe uno sin completar, el administrador podrá visualizar mensajes emergentes donde falten campos por llenar.</p>
<p>Resultado reflejado 1</p> 	
<p>Resultados Esperados 2</p>	<p>Si el administrador en el campo “Foto” ingreso un archivo que sea diferente a formato imagen, se le presentara un mensaje emergente “Error al cargar la foto, el archivo debe ser de tipo imagen”</p>
<p>Resultado reflejado 2</p> 	
<p>Resultados Esperados 3</p>	<p>Si el administrador cumplió con todos los requisitos para editar un alumno, como llenar todos los campos y en el campo “foto” subir un archivo de tipo imagen, al presionar el botón editar se le presentar un mensaje “Alumno actualizado con éxito”.</p>
<p>Resultado reflejado 3</p> 	

Resultados Esperados 4	Si el administrador decide no editar la información de un alumno, una vez estando en el formulario presiona el botón “Atrás” y regresa a la pantalla principal de la sección alumnos.
Evaluación de la Prueba	SUPERADA
Responsable	Tester del Equipo de Trabajo (Christian Guevara)

Tabla 14. Caso de Prueba CP_003

# Caso de Prueba	CP_003		
# Historia de Usuario	HU03	Fecha	10/02/2021
Descripción	Tiene por objetivo la respectiva verificación hacia el administrador del prototipo, es decir si puede o no eliminar a cualquier alumno registrado en algún curso del sistema.		
Condiciones de Ejecución	El administrador presiona la opción “Eliminar” de la sección alumnos.		
Entradas	<ul style="list-style-type: none"> - Selecciona el alumno a eliminar - Presiona el botón “Sí, Eliminar” 		
Resultados Esperados 1	Si el administrador decide no eliminar el alumno seleccionado, presiona el botón “Atrás” vuelve a la pantalla principal de la sección “Alumnos”.		
Resultado reflejado 1			
Evaluación de la Prueba	SUPERADA		
Responsable	Tester del Equipo de Trabajo (Christian Guevara)		

Especificaciones generales

Se consideró el sprint 1 “Microservicio Usuario (Alumnos)” como el principal para representar el proceso de desarrollo que realizan todos los demás Sprints ya que pasan por el mismo proceso, por lo que no fue necesario repetir la información.

Resultados de la validación de la propuesta mediante Alfa de Cronbach

En este apartado se presentan los resultados obtenidos de la encuesta aplicada a los expertos seleccionados, se detalla 1 sola pregunta la cual se consideró la más importante de la encuesta:

Tabla 15. Tabulación de la encuesta, pregunta N° 4.

	Experto 1	Experto 2	Experto 3
¿Cree usted que la investigación realizada en este proyecto generará un impacto tecnológico en las futuras generaciones de la Universidad Técnica de Cotopaxi?	4	5	3

Fuente: Christian Guevara (Investigador)

Figura 7. Tabulación pregunta 4.



Fuente: Christian Guevara (Investigador)

Al comprobar que el primer experto con un 42 % equivalente a excelente, el segundo experto con un 33% equivalente a muy bueno y el tercer experto con un 25% equivalente a bueno, consideraron que la investigación generará un impacto tecnológico en las futuras generaciones de la Universidad Técnica de Cotopaxi

A continuación, se presenta los resultados obtenidos a través del análisis de fiabilidad, por medio de Alfa de Cronbach para lo cual se realizó el cálculo mediante su fórmula establecida, además de esto es importante mencionar que el instrumento de validación de la propuesta, es decir la encuesta está compuesta de **8 ítems**, evaluados por **tres expertos** en el tema, proporcionando así los datos requeridos para proceder con la aplicación del cálculo y análisis de fiabilidad, el cual se presenta a continuación:

Tabla 16. Tabla estadística del método de Alfa de Cronbach

Encuestados	1	2	3	4	5	6	7	8	Sumatoria
E1	3	3	3	4	5	5	3	4	30
E2	5	5	4	5	5	5	4	5	38
E3	4	2	5	3	4	3	4	5	30
Varianza	0.667	1.556	0.667	0.667	0.222	0.889	0.222	0.222	
Sumatoria de varianzas	5.111								
Varianza de la suma de los ítems	14.222								

Fuente: Christian Guevara (Investigador)

Formula Alfa de Cronbach:

$$\alpha = \frac{K}{K - 1} \left[1 - \frac{\sum S_i^2}{S_T^2} \right]$$

Donde:

k: Número de ítems del instrumento = **8**

$\sum_{i=1}^k S_i^2$: Sumatoria de las varianzas de los ítems = **5.111**

S_T^2 : Varianza total del instrumento = **14.222**

α : Coeficiente de confiabilidad del cuestionario = **0.73**

Tabla 17. Rango de confiabilidad

RANGO	CONFIABILIDAD
0.53 a menos	Confiabilidad nula
0.54 a 0.59	Confiabilidad baja
0.60 a 0.65	Confiable
0.66 a 0.71	Muy confiable
0.72 a 0.99	Excelente confiabilidad
1	Confiabilidad perfecta

Con el presente análisis y cálculo mediante el coeficiente de fiabilidad de Alfa de Cronbach, se completa la validación del instrumento, obteniendo un resultado final de **0.73**, esta cifra representa una excelente confiabilidad según el rango perteneciente al método.

11.1.15 Acrónimos

- R. F.: Requisitos Funcionales
- R. N. F: Requisitos no Funcionales
- HU: Historias de Usuario
- CP: Casos de Prueba
- DoD: definición de hecho

12. IMPACTOS (TÉCNICOS, SOCIALES, AMBIENTALES O ECONÓMICOS)

12.1 Impacto Técnico

Para el desarrollo del presente prototipo se ha utilizado la metodología MicroIoT adaptándola a nuestra necesidad para el desarrollo del sistema, utilizando tecnologías actuales las cuales han resaltado en el campo tecnológico en los últimos años, provocando así un gran impacto tecnológico en los futuros estudiantes de la universidad técnica de Cotopaxi, ya que se presentan herramientas Open Source que se proyectan como a futuro para utilizarse con más frecuencia que actualmente, mediante las cuales se ha desarrollado este proyecto.

12.2 Impacto Social

Al realizar el prototipo como ejemplo para demostrar como los microservicios se desenvuelven trabajando de manera independiente pero a su vez en conjunto para llevar a cabo una tarea en especifica más grande, se crea oportunidad a futuros estudiantes y desarrolladores al ahorrar tiempo y esfuerzo ya que con el prototipo se les facilitara la comprensión del tema de manera significativa, con esto se fomenta el interés por la investigación del mismo tema o afines, teniendo como resultado mayor investigación, lo que generará que los estudiantes realicen sistemas mucho mejores y más completos que el prototipo presentado en esta investigación y como resultado final tengan reconocimiento ya sea dentro o fuera de la Universidad.

12.3 Impacto Económico

Se ha determinado que con la realización de esta investigación y prototipo a futuro se podría asegurar un impacto económico considerable, puesto que al inmiscuirse en tas nuevas formas de desarrollo de software, aprenderlas y ejecutarlas, se pueden conseguir resultados importantes para quienes se decidan aprender este tipo de arquitecturas ya que con su utilización al ser algo novedoso y que en los últimos años ha revolucionado el mercado tecnológico, se pueden generar grandes ingresos económicos, pues cada vez más empresas optan por contratar personal con experiencia en desarrollo de software y con conocimientos en arquitectura de microservicios, y mediante el desarrollo de este prototipo pueden usarlo como herramienta de aprendizaje.

13. PRESUPUESTO GENERAL

En este apartado se detallan los gastos del proyecto de investigación, en el cual se presentan cantidades reales entre gastos directos e indirectos.

13.1 Gastos directos

Tabla 18. Estimación de costos – Gatos Directos

Recursos	Cantidad	Valor Unitario, USD \$	Total, USD \$
Internet	1	33.60	33.60
Cuadernos	1	2.00	2.00
Esferos	3	0.50	1.50
Energía Eléctrica	1	20.00	20.00
Total, USD \$			57.10

Fuente: Christian Guevara (Investigador)

13.2 Gastos indirectos

Tabla 19. Estimación de costos – Gatos Indirectos

Detalle	Días	Valor Unitario, USD \$	Total, USD \$
Alimentación	140	1.5	210.00
Comunicación (Teléfono)	140	0.5	70.00
Total, USD \$			280.00

Fuente: Christian Guevara (Investigador)

13.3 Gastos generales

Tabla 20. Estimación de costos – Gastos Generales

Gastos Directos	57.10
Gastos Indirectos	280.00
Gastos Totales	337.10
Gastos Imprevistos 12%	40.45
Total	377.55

Fuente: Christian Guevara (Investigador)

13.4 Estimación del costo del software por puntos de historia

Para realizar el cálculo del costo del desarrollo del software correspondiente al prototipo desarrollado en esta investigación se tomando como herramienta principal el método de puntos de historia, este método se utiliza con el propósito de conseguir un valor en horas de esfuerzo para el desarrollo de una historia de usuario, a continuación, se presenta la realización del cálculo correspondiente.

Calculo:

$$Pph = \frac{(s/20)}{hl}$$

Donde:

S = Sueldo del desarrollador o programador

20 = Días laborables de cada mes

HL = Horas laborables

$$CD = (tdh * ht) * pph$$

Donde:

CD = Costo o valor del desarrollo

TDH = Total días de las historias de usuario completas

HT = Horas trabajadas

PPH = Precio por hora

Reemplazando resulta:

$$\text{PPH} = (400 / 20) / 8$$

$$\text{PPH} = 2.50$$

$$\text{CD} = (48 * 6) * 2.50$$

$$\text{CD} = \$720$$

Después del cálculo realizado se obtuvo un valor de 720 dólares, esto como resultado total del costo de desarrollo de software.

14. CONCLUSIONES Y RECOMENDACIONES

14.1. Conclusiones

- Se concluye que por medio de la revisión bibliográfica se pudo elaborar una estrategia metodológica para el desarrollo de microservicios mediante la cual se consiguió desarrollar un prototipo que tiene el objetivo de demostrar el trabajo de los microservicios y sus beneficios.
- Concluyo que mediante el estudio bibliográfico realizado se pudo determinar a MicroIoT como una metodología acorde para el desarrollo de microservicios, resaltando su utilización como tendencia a futuro en la industria del software y en los estudiantes de la Universidad Técnica de Cotopaxi.
- Se logro proponer una estrategia metodológica para el desarrollo de microservicios en la cual intervienen atributos de calidad, mismos que fueron establecidos e implementados en el prototipo final.
- Se concluye que producto de la revisión y estudio bibliográfico, además de la propuesta metodológica diseñada se pudo desarrollar un prototipo, el cual contiene varios microservicios interactuando entre sí para demostrar su desempeño.

14.2. Recomendaciones

- Recolectar información sobre futuras metodologías que puedan aparecer con el pasar de los años de fuentes indexadas que estén certificadas como revistas científicas, libros,

tesis, etc., ya que estas proporcionan datos verdaderos y sustentables, con datos que son fundamentales para el correcto desarrollo de una investigación.

- Se recomienda investigar más metodologías que tengan relación directa con microservicios, con la finalidad de fortalecer el conocimiento acerca de esta arquitectura y que sirva como herramienta para adentrarse más en el mundo del desarrollo de software.
- Recomiendo reforzar la investigación bibliográfica sobre atributos de calidad, con el propósito de comparar el avance tecnológico que seguramente vaya a tener el desarrollo de microservicios.
- Se recomienda utilizar el prototipo como herramienta de aprendizaje sobre la arquitectura de microservicios, a fin de que en un futuro los estudiantes de la Universidad técnica de Cotopaxi lo tomen como punto de partida para el desarrollo de un modelo más completo y posteriormente una posible implementación.

15. BIBLIOGRAFÍA

-chile, S., & Arévalo del Río, F. (2016). *UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA DEPARTAMENTO DE INFORMÁTICA El enfoque de microservicios como estrategia para mejorar la calidad del software.*

<https://repositorio.usm.cl/bitstream/handle/11673/24328/3560902038531UTFSM.pdf?sequence=1&isAllowed=y>

Aguirre, G. (2018). *Universidad técnica del norte facultad de ingeniería en ciencias aplicadas carrera de ingeniería en electrónica y redes de comunicación evaluación 1 (1. 1, 149. [http://repositorio.utn.edu.ec/bitstream/123456789/7979/1/04 MEC 213 TRABAJO DE GRADO.pdf](http://repositorio.utn.edu.ec/bitstream/123456789/7979/1/04_MEC_213_TRABAJO_DE_GRADO.pdf)*

Alfonso, D., & Contreras, B. (2018). Arquitectura de Microservicios. *Tecnología Investigación y Academia*, 6(1), 36–46.

Arboleda Cola Carlos Augusto. (2017). *Propuesta Metodológica Para Migración De Sistemas Web Con Arquitectura Monolítica Hacia Una Arquitectura Basada En Microservicios.* file:///C:/Users/miguel/Downloads/CD-8386.pdf

Cabrera Alvarado, E. F., & Cárdenas Cárdenas, P. J. (2018). Metodología para la creación de aplicaciones basadas en Microservicios para soluciones de Internet de las Cosas en

- Ambientes de Vida Asistidos. *UNIVERSIDAD DE CUENCA*, 1–300.
<http://dspace.ucuenca.edu.ec/handle/123456789/31511>
- Cataldi, Z, Lage, F, Pessacq, R. (1997). *Ingenieria de software educativo*.
- Cojocar, M., Uta, A., & Oprescu, A. M. (2019). MicroValid: A validation framework for automatically decomposed microservices. *Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom, 2019-Decem*, 78–86.
<https://doi.org/10.1109/CloudCom.2019.00023>
- De la Cruz Vélez de Villa, P. E., Espinoza Ramirez, M. H., & Cuba Estrella, O. (2019). Propuesta de arquitectura de microservicios, metodología Scrum para una aplicación móvil de control académico: Caso Escuela Profesional de Obstetricia de la UNMSM. *Hamut' Ay*, 6(2), 141–158. <https://doi.org/10.21503/hamu.v6i2.1781>
- Deemer, P., Larman, C., Vodde, B., & Benefield, G. (2012). Una introducción básica a la teoría y práctica de Scrum. *InfoQ*, 20.
https://scrumprimer.org/primers/es_scrumprimer20.pdf
- DÍAZ, A. G. (n.d.). *INGENIERÍA DE SOFTWARE AVANZADA*. 1–51.
- Ginestà, Marc Gibert, O. P. M. (2012). *Bases de datos en PostgreSQL*. 1–21.
- Gudiño, Q. A. R. (2018). Estudio De Integración De Los Frameworks Angular 4 Y Yii2 , Orientado a Servicios Rest , Que Permitan La Gestión Y Control De Inventarios Para Mejorar La Productividad En La Empresa Induxion. *Universidad Técnica Del Norte*.
- Hernández, M., Cantero, P., & Delgado, T. (2011). *Desarrollo de una aplicación web para la gestión de entornos virtuales*. 83. <http://eprints.ucm.es/13083/>
- Jazmine Escobar-Pérez, Á. C.-M. (2008). VALIDEZ DE CONTENIDO Y JUICIO DE EXPERTOS: UNA APROXIMACIÓN A SU UTILIZACIÓN. *Polymer*, 19(2), 231–233.
[https://doi.org/10.1016/0032-3861\(78\)90049-6](https://doi.org/10.1016/0032-3861(78)90049-6)
- Kenneth Calvo, Johan Durán, Esteban Quirós, E. M. (2017). MongoDB : alternativas de implementar y consultar documentos. *Universidad de Costa Rica, Costa Rica*.
- Lewis, J., & Fowler, M. (n.d.). *Microservices*.
<https://Martinowler.Com/Articles/Microservices.Html>.
- Marco, G. J. (2016). *DevOps , la nueva tendencia en el desarrollo de sistemas TI , un caso*

- práctico en el análisis de incidencias de software Control Documental*. 102.
[https://upcommons.upc.edu/bitstream/handle/2117/85074/Análisis de incidentes en el ámbito TIC con DevOps.pdf?sequence=1&isAllowed=y](https://upcommons.upc.edu/bitstream/handle/2117/85074/Análisis%20de%20incidentes%20en%20el%20ámbito%20TIC%20con%20DevOps.pdf?sequence=1&isAllowed=y)
- Marqués, M. (2009). *Bases de Datos*. 227.
http://www3.uji.es/~mmarques/apuntes_bbdd/apuntes.pdf
- Maya, E., y López, D. (2017). Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web. *Septima Conferencia de Directores de Tecnología de Información 2017, July*, 12. [http://dspace.redclara.net:8080/bitstream/10786/1277/1/93Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web.pdf](http://dspace.redclara.net:8080/bitstream/10786/1277/1/93Arquitectura%20de%20Software%20basada%20en%20Microservicios%20para%20Desarrollo%20de%20Aplicaciones%20Web.pdf)
- Menzinsky, Alexander, López, Gertrudis, Palacio, J. (2018). *Historias de Usuario*.
- Muñoz Hernández, R. E. (2018). *Investigación y desarrollo de la arquitectura orientada a microservicios para la implementación de módulos en un sistema web*. 26–126.
- Müssig, D., Stricker, R., Lassig, J., & Heider, J. (2017). Highly scalable microservice-based enterprise architecture for smart ecosystems in hybrid cloud environments. *ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems*, 3(Iceis), 454–459. <https://doi.org/10.5220/0006373304540459>
- Navarrete, T. (2006). *El lenguaje JavaScript*.
- Pareja Valerio, C. B., & Burgos Robles, L. J. (2019). *La arquitectura de software basada en microservicios: Una revisión sistemática de la literatura*. 1–23.
<papers2://publication/uuid/45D7E632-B571-4218-9E47-8B4457FEA9D3>
- Pérez, J. E. (2019). *Introducción a JavaScript*.
- Quevedo-Avila, P., Zhindón-Mora, M. G., & Quevedo-Sacoto, A. S. (2020). *Arquitectura de microservicios para compras en línea : caso de uso "ala orden" Microservices architecture for online shopping : "to order" use case Arquitetura de microsserviços para compras online : caso de uso "sob encomenda"*. 5(1), 151–162.
<https://doi.org/10.23857/pc.v5i1.1884>
- Ramon, Z. V. (2020). *Desarrollo de aplicaciones web utilizando Angular como framework*.
- Retamal Valenzuela, J. H. (2019). *Plataforma de desarrollo de aplicaciones en el dcc basada*

en técnicas de devops.

- Rivadeneira, A. H. (2019). Desarrollo ágil del nuevo sistema institucional basado en una arquitectura orientada a microservicios. *Universidad de Montemorelos Facultad*, 1(1), 1–8.
<http://dx.doi.org/10.1016/j.cirp.2016.06.001><http://dx.doi.org/10.1016/j.powtec.2016.12.055><https://doi.org/10.1016/j.ijfatigue.2019.02.006><https://doi.org/10.1016/j.matlet.2019.04.024><https://doi.org/10.1016/j.matlet.2019.127252><http://dx.doi.org/10.1016/j.cirp.2016.06.001>
- Rivera caro, e. h. (2019). Universidad Nacional Tecnológica de Lima sur. *implementación de una aplicación móvil en android para mostrar los atractivos turísticos de la reserva nacional de lachay, aplicando la metodología scrum y herramientas devops*, 1–177.
<http://repositorio.untels.edu.pe/handle/untels/166>
- Saavedra, A. V. (2019). *Interfaz gráfica de usuario para una plataforma de máquinas industriales sensorizadas.*
- Santiago Alejandro García Vela. (2016). Desarrollo de un Sistema web para el análisis y monitoreo estadístico de los datos que genera el pi-ssan en las comunidades rurales de Chimborazo -Ecuador. *Escuela Superior Politécnica de Chimborazo.*
- Seneque, N. R. (2014). Administración en MongoDB. *Universidad de Valladolid.*
<http://uvadoc.uva.es/handle/10324/7898>
- Valdivia, J. A., Limon, X., & Cortes-Verdin, K. (2019). Quality attributes in patterns related to microservice architecture: A Systematic Literature Review. *Proceedings - 2019 7th International Conference in Software Engineering Research and Innovation, CONISOFT 2019*, 181–190. <https://doi.org/10.1109/CONISOFT.2019.00034>
- Vera-Rivera, F. H. (2018). Método de automatización del despliegue continuo en la nube para la implementación de microservicios. *Avances En Ingeniería de Software a Nivel Iberoamericano, CibSE 2018*, 597–604.
- Verdier, D., & Rodriguez, G. (2020). *Implementación de Patrones de Microservicios.* 5–129.
- Zafra Ramirez, I. K. (2020). E-commerce basado en microservicios para el proceso de ventas en la empresa COMATPERU S.A.C. *Universidad Cesar Vallejo*, 186.
http://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/47102/Gutierrez_RS-SD.pdf?sequence=1&isAllowed=y

16. ANEXOS

16.1 Instrumento utilizado para la validación de la propuesta.

En este apartado se presenta la encuesta, instrumento utilizado para validar la propuesta mediante juicio de expertos.

16.2 Encuesta

Validación y Verificación de los Requerimientos Implementados

Es necesario que responda las siguientes preguntas de una forma muy objetiva.

En una escala del 1 al 5 cuál cree que es la factibilidad o la utilidad de la siguiente temática: Estudio metodológico para el desarrollo de microservicios, diseño de un prototipo y a la vez permita organizar por orden de prioridad estos conocimientos siendo 1 considerado como **malo** y 5 considerado como **muy bueno**.

N.º	Pregunta	Ponderación				
		1	2	3	4	5
1	¿Cree usted que es necesario considerar la utilización de la arquitectura de microservicios como tendencia portadora de futuro en el desarrollo de software?					
2	¿Para esta investigación cree conveniente utilizar una metodología estrictamente relacionada con el desarrollo de microservicios?					
3	¿Considera apropiado la utilización de varias bases de datos para conectar diferentes microservicios y así demostrar su desempeño?					
4	¿Cree usted que la investigación realizada en este proyecto generará un impacto tecnológico en las futuras generaciones de la Universidad Técnica de Cotopaxi?					
5	¿Cree usted apropiado el desarrollo de un prototipo para reforzar los conocimientos sobre la arquitectura de microservicios?					
6	¿Considera necesario realizar investigaciones relacionadas a nuevas tecnologías como son los microservicios para elevar el nivel de conocimiento sobre el desarrollo de software?					
7	¿En cuestión de seguridad cree usted conveniente utilizar varias bases de datos para una misma aplicación?					
8	¿Cree usted necesario establecer una comparación entre arquitecturas tradicionales y la arquitectura de microservicios, con el fin de establecer ventajas y desventajas?					

16.3 Cartas dirigidas a los expertos

A continuación, se presentan las cartas dirigidas a los 3 expertos con sus respectivas firmas, los cuales ayudaron con la aplicación de la encuesta, misma que se utilizó como instrumento de recolección de datos para poder determinar la confiabilidad que tiene el proyecto mediante el método de Juicio de Expertos.

UNIVERSIDAD TÉCNICA DE COTOPAXI

Proyecto de Investigación

Nombre del Experto: Edwin Bladimir Casa Sivinta

El siguiente cuestionario va dirigido a usted sabiendo que es un experto en la especialización de sistemas de información y temas relacionados con microservicios, me dirijo hacia usted para solicitarle su colaboración en calidad de posible experto en el desarrollo de software.

Si está dispuesto usted a ofreceme su ayuda se necesita antes de consultarlo determinar su coeficiente de competencia en el tema antes planteado para poder reforzar la validez del resultado planteado.

Datos del experto

Nombres y Apellidos:	Casa Sivinta Edwin Bladimir
Grado Académico:	Tercer Nivel
Lugar de Trabajo:	KreandoTI
Cargo que Desempeña:	Desarrollador de Software
Años de Experiencia:	2



Casa Sivinta Edwin Bladimir

UNIVERSIDAD TÉCNICA DE COTOPAXI

Proyecto de Investigación

Nombre del Experto: Ing. Robinson Yáñez

El siguiente cuestionario va dirigido a usted sabiendo que es un experto en la especialización de sistemas de información y temas relacionados con microservicios, me dirijo hacia usted para solicitarle su colaboración en calidad de posible experto en el desarrollo de software.

Si está dispuesto usted a ofrecerme su ayuda se necesita antes de consultarlo determinar su coeficiente de competencia en el tema antes planteado para poder reforzar la validez del resultado planteado.

Datos del experto

Nombres y Apellidos:	Robinson Andrés Yáñez Núñez.
Grado Académico:	Tercer Nivel.
Lugar de Trabajo:	CASRAY Software Solutions.
Cargo que Desempeña:	Desarrollador de Software.
Años de Experiencia:	2



Ing. Robinson Yáñez

UNIVERSIDAD TÉCNICA DE COTOPAXI

Proyecto de Investigación

Nombre del Experto: Yunda Cando Oscar Javier

El siguiente cuestionario va dirigido a usted sabiendo que es un experto en la especialización de sistemas de información y temas relacionados con microservicios, me dirijo hacia usted para solicitarle su colaboración en calidad de posible experto en el desarrollo de software.

Si está dispuesto usted a ofreceme su ayuda se necesita antes de consultarlo determinar su coeficiente de competencia en el tema antes planteado para poder reforzar la validez del resultado planteado.

Datos del experto

Nombres y Apellidos:	Oscar Javier Yunda Cando
Grado Académico:	Tercer Nivel
Lugar de Trabajo:	KreandoTI
Cargo que Desempeña:	Analista Desarrollador de Software
Años de Experiencia:	2



Yunda Cando Oscar Javier