



UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
SISTEMAS DE INFORMACIÓN

PROYECTO DE INVESTIGACIÓN

Tema: Diseño e implementación de aplicaciones basadas en microservicios utilizando un modelo de optimización de dependencia y complejidad.

Proyecto de Titulación presentado previo la obtención del Título de Ingeniero en Sistemas de Información.

Autores:

Salazar Ninasunta Alexandra Jeaneth

Toasa Supe María José

Tutor Académico:

Ing. Mg. Tapia Cerda Verónica del Consuelo

LATACUNGA-ECUADOR

Abril – Agosto 2022

DECLARATORIA DE AUTORÍA

Nosotras, **SALAZAR NINASUNTA ALEXANDRA JEANETH**, con número de C.I: **055000759-5**, y **TOASA SUPE MARÍA JOSÉ** con número de C.I: **180466481-9**, declaramos ser autoras del presente proyecto de investigación: “**DISEÑO E IMPLEMENTACIÓN DE APLICACIONES BASADAS EN MICROSERVICIOS UTILIZANDO UN MODELO DE OPTIMIZACIÓN DE DEPENDENCIA Y COMPLEJIDAD**”, siendo la **Ing. MSc. VERÓNICA DEL CONSUELO TAPIA CERDA**, tutora del presente trabajo de investigación; y eximamos expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

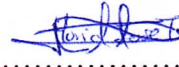
Además, certificamos que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de nuestra exclusiva responsabilidad.

Atentamente



.....
Salazar Ninasunta Alexandra Jeaneth

CI: 055000759-5



.....
Toasa Supe María José

CI: 180466481-9



AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN

En calidad de Tutor del Trabajo de Investigación sobre el título: “DISEÑO E IMPLEMENTACIÓN DE APLICACIONES BASADAS EN MICROSERVICIOS UTILIZANDO UN MODELO DE OPTIMIZACIÓN DE DEPENDENCIA Y COMPLEJIDAD”, de SALAZAR NINASUNTA ALEXANDRA JEANETH con C.I. 0550007595 Y TOASA SUPE MARÍA JOSÉ con C.I.1804664819, de la carrera de INGENIERÍA EN SISTEMAS DE INFORMACIÓN, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Consejo Directivo de la Facultad de CIENCIAS DE LA INGENIERÍA Y APLICADAS de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

Latacunga, Agosto, 2022

A handwritten signature in blue ink, appearing to be 'Verónica del Consuelo', written over a horizontal line.

Tutor de Titulación

Ing. Mg. Tapia Cerda Verónica del Consuelo

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, las postulantes: **Salazar Ninasunta Alexandra Jeaneth** con **C.I. 0550007595** y **Toasa Supe María José** con **C.I. 1804664819**, con el título de Proyecto de titulación: **“DISEÑO E IMPLEMENTACIÓN DE APLICACIONES BASADAS EN MICROSERVICIOS UTILIZANDO UN MODELO DE OPTIMIZACIÓN DE DEPENDENCIA Y COMPLEJIDAD”**, han 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, Agosto 2022

Para constancia firman:

Lector 1 (Presidente)
Nombre: Mg. Quinatoa Arequipa Edwin Edison
CC: 050256337-2

Lector 2
Nombre: Mg. Iza Carate Miryan Dorila
CC: 050195761-7

Lector 3
Nombre: Dr. Cadena Moreano José Augusto
CC: 050155279-8

AGRADECIMIENTO

Agradezco primeramente a Dios todo poderoso, por brindarme salud y vida para poder cumplir mis sueños y metas, en segundo lugar, un agradecimiento incondicional a mi esposo Fernando C. y a mis padres Rafael S. y Ofelia N. quienes estuvieron conmigo en todo momento apoyándome y dándome fuerzas para culminar este largo camino.

Agradezco a todos los docentes que formaron parte de mi vida universitaria, y especialmente a la Ing. Mg. Verónica Tapia por ser una excelente profesional, que nos guió y nos brindó su ayuda compartiendo su conocimiento durante el desarrollo del proyecto de investigación y así cumplir la meta que me he propuesto.

Un agradecimiento sincero a mi familia y amigos que me han aconsejado y me apoyaron para nunca darme por vencida y culminar este trabajo de titulación.

Alexandra Salazar

DEDICATORIA

Este trabajo de investigación se lo dedico a Dios, a mi esposo Fernando, a mi hija Montserrat y a mis queridos padres Rafael y Ofelia, que han sido mi fuerza en momentos difíciles como en los momentos buenos, impulsándome para cumplir mi sueño de ser una profesional.

A mis hermanos Daniel, Mayra, Erika y Tito que siempre han estado conmigo ayudándome con sus palabras de aliento que no me dejaba darme por vencida y seguir adelante luchando por mis sueños y esforzándome para ser mejor cada día.

Alexandra Salazar

AGRADECIMIENTO

Mi agradecimiento primeramente a Dios, por derramar de sus bendiciones sobre mí, a mi esposo, a mis padres, a mis hijos quienes siempre me han enseñado que con perseverancia y dedicación los sueños se pueden hacer realidad.

Agradezco la Universidad Técnica de Cotopaxi por formarme académicamente junto a su distinguido grupo de profesionales, a mi tutora de tesis, Ing. Mg. Verónica Tapia por compartir sus conocimientos, su tiempo mientras desarrollaba el proyecto de investigación y así cumplir la meta que me he propuesto.

Un agradecimiento sincero a mis hermanas y sobrino quienes han estado en todo momento brindándome su apoyo sin importar la distancia.

María José Toasa Supe

DEDICATORIA

Dedico este proyecto de investigación con gratitud y amor eterno a mi Dios, porque sin Él esto no hubiera sido posible, me dio la oportunidad de alcanzar una más de mis metas y se con certeza que el seguirá derramando de sus bendiciones sobre mí.

A mí amada madre Fabiola S. siempre me ha impulsado a ser mejor me ha brindado su apoyo incondicional, a pesar de la distancia, sé que este momento es tan especial para ti como lo es para mí. A mi amado padre Ángel T. cuyas enseñanzas y sabios consejos me han ayudado a ser perseverante en la búsqueda de mis sueños.

A mi amado esposo Francisco V. quien me ha brindado el apoyo constante para alcanzar esta meta, aunque hemos pasado momentos difíciles siempre has estado apoyándome. A mis tesoros Esther, Santiago y Sara quienes han sido mi fuente de motivación para superarme cada día.

A mis amadas hermanas Johanna, Rebeca y mi sobrino Isaac en todo momento me han brindado su apoyo y cariño.

María José Toasa Supe

ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA.....	ii
AVAL DEL TUTOR DEL PROYECTO DE TITULACIÓN.....	iii
APROBACIÓN DEL TRIBUNAL DE TITULACIÓN.....	iv
AGRADECIMIENTO	v
DEDICATORIA.....	vi
AGRADECIMIENTO	vii
DEDICATORIA.....	viii
ÍNDICE GENERAL	ix
ÍNDICE DE TABLAS.....	xiv
ÍNDICE DE FIGURAS	xv
ÍNDICE DE ECUACIONES	xvi
RESUMEN.....	xvii
ABSTRACT	xviii
AVAL DE TRADUCCIÓN.....	xix
1. INFORMACIÓN GENERAL	xx
2. INTRODUCCIÓN.....	1
2.1. EL PROBLEMA.....	2
2.1.1. Situación Problémica.....	2
2.1.2. Formulación del problema	2
2.2. OBJETO Y CAMPO DE ACCIÓN	3
2.3. BENEFICIARIOS.....	3
2.4. JUSTIFICACIÓN	3
2.5. HIPÓTESIS.....	4
2.6. OBJETIVOS	4
2.6.1. General:	4
2.6.2. Específicos:	4

2.7. SISTEMA DE TAREAS.....	5
3. FUNDAMENTACIÓN TEÓRICA	6
3.1. ANTECEDENTES	6
3.2. APLICACIONES WEB.....	7
3.3. INGENIERÍA DE SOFTWARE.....	7
3.3.1. Modelos de Proceso de Software	7
3.3.2. Modelo Clásico	8
3.3.3. Modelos de proceso incremental	9
3.3.4. Modelo Espiral	9
3.3. METODOLOGÍAS TRADICIONALES.....	9
3.4. METODOLOGÍAS ÁGILES.....	10
3.4.1. Scrum	11
3.4.2. Método de MosCoW	13
3.4.3. Release planning	13
3.4.4. Planinig Poker	13
3.4.5. DevOps.....	13
3.5. CALIDAD DE SOFTWARE.....	14
3.5.1. Clasificación de la calidad del software.....	15
3.5.2. Norma ISO 9126	16
3.5.3. Calidad en microservicios	17
3.6. ARQUITECTURAS DE SOFTWARE	20
3.6.1. Arquitectura de software monolítica.....	20
3.6.2. Arquitecturas de software basadas Microservicios	20
3.7. HERRAMIENTAS DE DESARROLLO	22
3.7.1. Backend.....	22
3.7.2. Frontend	25
3.8. CÁLCULO DE LA COMPLEJIDAD Y LA DEPENDENCIA EN LA	

ARQUITECTURA BASADA EN MICROSERVICIOS	26
3.8.1. Fórmulas para el cálculo de la dependencia.....	26
3.8.2. Fórmulas para calcular la complejidad.....	30
3.9. JUICIO DE EXPERTOS	34
3.10. COEFICIENTE ALFA DE CRONBACH.....	35
3.11. METODOLOGÍAS DE INVESTIGACIÓN	35
3.11.1. Concepto de investigación	35
3.11.2. Enfoque Cuantitativo.....	35
3.11.3. Enfoque Cualitativo.....	36
4. MATERIALES Y MÉTODOS.....	36
4.1. TIPO DE INVESTIGACIÓN	36
4.1.1. Investigación cuali-cuantitativa.....	36
4.2. NIVEL DE INVESTIGACIÓN	36
4.2.1. Investigación descriptiva.....	36
4.3. DISEÑO DE INVESTIGACIÓN	36
4.3.1. Diseño narrativo	36
4.4. INSTRUMENTOS DE RECOLECCIÓN DE DATOS.....	37
4.4.1. Fichas nemotécnicas.....	37
4.4.2. Encuesta	37
4.5. TÉCNICA DE ANALISIS DE DATOS	37
4.5.1. Juicios de expertos	37
4.6. TÉCNICAS DE DESARROLLO DE SOFTWARE	37
4.6.1. Historias de usuario.....	37
4.6.2. Product Backlog	38
4.6.3. Release Planning	38
4.6.4. Sprint	38
4.6.5. Técnica de MoSCoW	38

4.7. MODELO DE OPTIMIZACIÓN DE DEPENDENCIA Y COMPLEJIDAD	38
4.7.1. Acoplamiento de MSBA	38
4.7.2. Complejidad cognitiva de MSBA	39
5. ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS	40
5.1. MODELO DE OPTIMIZACIÓN PROPUESTO	40
5.2. APLICACIÓN DE LAS DISTINTAS PROPUESTAS DE CÁLCULO DE DEPENDENCIA Y COMPLEJIDAD PARA EL CASO DE ESTUDIO	42
5.2.1. Cálculo de la dependencia.....	42
5.2.2. Cálculo de la Complejidad	49
5.3. COMPARACIÓN DE LOS RESULTADOS DE LAS PROPUESTAS PARA EL CÁLCULO DE DEPENDENCIA Y COMPLEJIDAD.....	55
5.4. DISEÑO DEL MODELO ARQUITECTÓNICO.....	56
5.5. IMPLEMENTACIÓN.....	57
5.5.1. Historias de usuario.....	57
5.5.2. Product Backlog de la aplicación	59
5.5.3. Asignación de las historias de usuario	60
5.5.4. Base de datos por servicio.....	62
5.5.5. Ejecución del sprint.....	63
5.6. VALIDACIÓN	74
5.6.1. Resultados de la validación de la propuesta mediante el coeficiente Alfa de Cronbach	75
5.7. IMPACTOS (TÉCNICOS, SOCIALES, AMBIENTALES O ECONÓMICOS)	84
5.7.1. Impacto Técnico.....	84
5.7.2. Impacto Social.....	84
5.7.3. Impacto económico	85
5.8. PRESUPUESTO GENERAL	85
5.8.1. Gastos directos	85
5.8.2. Gastos indirectos	86

5.8.3. Estimación del costo del software por puntos de historia	86
5.8.4. Gastos generales.....	87
6. CONCLUSIONES Y RECOMENDACIONES	88
6.1. CONCLUSIONES	88
6.2. RECOMENDACIONES.....	88
7. BIBLIOGRAFÍA.....	89
8. ANEXOS	114
ANEXO 1: INFORME DE PLAGIO	114
ANEXO 2: ARQUITECTURA DEL PROYECTO	115
ANEXO 3: TABLA DE RESULTADOS.....	155
ANEXO 4: DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	157
ANEXO 5: CARTA A EXPERTOS PARA EVALUACIÓN DE LA ENCUESTA	158
ANEXO 6: HOJA DE VIDA DEL EXPERTO	161

ÍNDICE DE TABLAS

Tabla 1. Beneficiarios directos e indirectos.....	3
Tabla 2. Actividades y tareas.....	5
Tabla 3. Aspectos relevantes entre metodología tradicional vs metodología ágil.....	11
Tabla 4. Estándares y atributos de calidad.....	15
Tabla 5. Comparación de las métricas calculadas	56
Tabla 6. Historias de usuario	58
Tabla 7. Priorización del Product Backlog	59
Tabla 8. Asignación de dependencia según las historias de usuario por microservicio	60
Tabla 9. Historia de Usuario HU006 - Crear curso	64
Tabla 10. Historia de Usuario HU007 - Mostrar cursos	64
Tabla 11. Historia de Usuario HU008 - Modificar curso	65
Tabla 12. Historia de Usuario HU009 - Dar de baja curso.....	66
Tabla 13. Historia de Usuario HU012 - Crear examen	67
Tabla 14. Historia de Usuario HU013 - Mostrar exámenes	67
Tabla 15. Casos de Prueba N 006 - Crear curso	69
Tabla 16. Casos de Prueba N 007 - Mostrar cursos.....	70
Tabla 17. Casos de Prueba N 008 - Modificar cursos	70
Tabla 18. Casos de Prueba N 009 - Dar de baja alumnos.....	71
Tabla 19. Casos de Prueba N 012 - Crear examen	72
Tabla 20. Casos de Prueba N 013 – Mostrar exámenes	74
Tabla 21. Resultados del instrumento de validación aplicado a tres expertos.....	75
Tabla 22. Tabulación de la encuesta, pregunta N° 1	76
Tabla 23. Tabulación de la encuesta, pregunta N° 2	77
Tabla 24. Tabulación de la encuesta, pregunta N° 3	78
Tabla 25. Tabulación de la encuesta, pregunta N° 4	79
Tabla 26. Tabulación de la encuesta, pregunta N° 5	80
Tabla 27. Tabulación de la encuesta, pregunta N° 6	81
Tabla 28. Tabulación de la encuesta, pregunta N° 7	82
Tabla 29. Tabla estadística del método de Alfa de Cronbach	83
Tabla 30. Rango de confiabilidad	84
Tabla 31. Estimación de costos – Gastos Directos	85
Tabla 32. Estimación de costos - Gastos indirectos	86
Tabla 33. Total de gastos	87

ÍNDICE DE FIGURAS

Figura 1. Tipos de enfoques según [5]	8
Figura 2. Etapas Modelo Clásico según [6].....	8
Figura 3. Flujo Procesos lineales según [7].....	9
Figura 4. Flujo Procesos lineales según [7].....	11
Figura 5. Arquitectura Monolítica según [10].....	20
Figura 6. Esquema del modelo de optimización -Dependencia y complejidad	41
Figura 7. Modelo del diseño de la aplicación.....	57
Figura 8. Microservicios, asignación de historias de usuario.....	61
Figura 9. Calls y Request.....	62
Figura 10. Base de datos por servicio.....	63
Figura 11. Tabulación pregunta 1	76
Figura 12. Tabulación pregunta 2.....	77
Figura 13. Tabulación pregunta 3.....	78
Figura 14. Tabulación pregunta 4.....	79
Figura 15. Tabulación pregunta 5.....	80
Figura 16. Tabulación pregunta 6.....	81
Figura 17. Tabulación pregunta 7.....	82

ÍNDICE DE ECUACIONES

Ecuación 1. Importancia absoluta del microservicio	26
Ecuación 2. Importancia absoluta total del microservicio.....	26
Ecuación 3. Dependencia absoluta del microservicio	27
Ecuación 4. Dependencia absoluta total del microservicio	27
Ecuación 5. Interdependencia de los microservicios.....	27
Ecuación 6. Interdependencia total de los microservicios.....	27
Ecuación 7. Acoplamiento de los microservicios.....	27
Ecuación 8. Acoplamiento total de los microservicios.....	27
Ecuación 9. Falta de cohesión	28
Ecuación 10. Grado de falta de cohesión.....	28
Ecuación 11. Grado total de falta de cohesión	28
Ecuación 12. Acoplamiento.....	28
Ecuación 13. Cohesión	29
Ecuación 14. Llamadas y solicitudes.....	29
Ecuación 15. Acoplamiento total del microservicio.....	30
Ecuación 16. Llamadas de un microservicio	30
Ecuación 17. Puntos de historia de usuario del microservicio	31
Ecuación 18. Puntos de complejidad cognitiva de MSBA	32
Ecuación 19. Mayor granularidad.....	32
Ecuación 20. Valores del caso base.....	32
Ecuación 21. Caso base	32
Ecuación 22. Complejidad total del sistema.....	33
Ecuación 23. Granularidad de un servicio.....	33
Ecuación 24. Complejidad del servicio	33
Ecuación 25. Complejidad total del sistema.....	33
Ecuación 26. Complejidad estructural.....	34
Ecuación 27. Complejidad de datos.....	34
Ecuación 28. Complejidad total del sistema.....	34
Ecuación 29. Coeficiente Alfa de Cronbrach	35

RESUMEN

Título:

DISEÑO E IMPLEMENTACIÓN DE APLICACIONES BASADAS EN MICROSERVICIOS UTILIZANDO UN MODELO DE OPTIMIZACIÓN DE DEPENDENCIA Y COMPLEJIDAD.

Autores:

Salazar Ninasunta Alexandra Jeaneth

Toasa Supe María José

Tutor: Ing. Mg. Verónica Tapia Cerda

El presente proyecto de investigación se basó en una revisión bibliográfica que detalla la arquitectura basada en microservicios como un nuevo enfoque arquitectónico y organizativo en el desarrollo de software, donde las aplicaciones están divididas en pequeños servicios independientes y se comunican a través de mensajes más conocidos como APIS, además permitió resaltar el cambio que se ha producido en la industria de software al trabajar con un monolito y un microservicio. El objetivo principal de esta investigación ha sido proponer un modelo de optimización de dependencia y complejidad para el diseño de aplicaciones basadas en microservicios mediante la evaluación de distintas propuestas, el desarrollo del modelo arquitectónico se realizó utilizando el cálculo de las métricas del modelo de optimización y prácticas Ágiles que permitieron gestionar el proceso de diseño del caso de estudio. Para validar y verificar la hipótesis planteada se aplicó la técnica de juicio de expertos a través de un instrumento de validación, el análisis de fiabilidad del proyecto de investigación se realizó a través del método de coeficiente alfa de Cronbach.

Palabras claves: microservicios, arquitecturas, desarrollo de software, prácticas Ágiles, métricas de dependencia y complejidad.



FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
CARRERA DE SISTEMAS DE INFORMACIÓN

ABSTRACT

Topic:

APPLICATIONS DESIGN AND IMPLEMENTATION BASED ON
MICROSERVICES USING DEPENDENCE AND COMPLEXITY OPTIMIZATION
A MODEL.

Authors:

Salazar Ninasunta Alexandra Jeaneth

Toasa Supe María José

Tutor: Ing. Mg. Tapia Cerda Verónica Consuelo

The current research project was based on a bibliographical review that details the architecture based on microservices as a new architectural and organizational approach in the software development, where applications are divided into small independent services and communicate, through messages better known as APIS also, they made to highlight the change, which has occurred in the software industry by working with a monolith and a microservice. The main aim this research has been to propose a dependency and complexity optimization model for the applications design based on microservices, through the different proposals assessment, it was carried the architectural model development using the optimization model metrics calculation and Agile practices, which allowed managing the case study design process. To validate and verify the proposed hypothesis was applied the expert judgment technique, through a validation instrument, it was made the research project reliability analysis, through the Cronbach's alpha coefficient method.

Keywords: Microservices, architectures, software development, agile practices, dependency metrics and complexity.

AVAL DE TRADUCCIÓN

En calidad de Docente del Idioma Inglés del Centro de Idiomas de la Universidad Técnica de Cotopaxi; en forma legal **CERTIFICO** que:

La traducción del resumen al idioma Inglés del trabajo de titulación cuyo título versa: **“DISEÑO E IMPLEMENTACIÓN DE APLICACIONES BASADAS EN MICROSERVICIOS UTILIZANDO UN MODELO DE OPTIMIZACIÓN DE DEPENDENCIA Y COMPLEJIDAD.”** presentado por: **Salazar Ninasunta Alexandra Jeaneth y Toasa Supe María José**, estudiantes de la Carrera de: **Sistemas de Información**, perteneciente a la **Facultad de Ciencias de la Ingeniería y Aplicadas**, lo realizaron bajo mi supervisión y cumple con una correcta estructura gramatical del Idioma.

Es todo cuanto puedo certificar en honor a la verdad y autorizo a las peticionarias hacer uso del presente aval para los fines académicos legales.

Latacunga, septiembre del 2022

Atentamente,



Mg. Marco Beltrán



CENTRO
DE IDIOMAS

DOCENTE CENTRO DE IDIOMAS-UTC
CI: 0502666514

1. INFORMACIÓN GENERAL

Título: Diseño e implementación de aplicaciones basadas en microservicios utilizando un modelo de optimización de dependencia y complejidad.

Tipo de proyecto: Proyecto de investigación

Fecha de inicio: 25/10/2121

Fecha de finalización: 27/02/2022

Lugar de ejecución: Latacunga

Facultad que auspicia: Ciencias de la Ingeniería y Aplicadas

Carrera que auspicia: Sistemas de Información

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.

Equipo de trabajo:

Tutor de Titulación

Apellidos y Nombres: Ing. Mg. Tapia Cerda Verónica Consuelo

Cédula: 0502053697

Correo: veronica.tapia@utc.edu.ec

Teléfono: 0992952383

Estudiante

Apellidos y Nombres: Salazar Ninasunta Alexandra Jeaneth

Cédula: 0550007595

Correo: alexandra.salazar7595@utc.edu.ec

Teléfono: 0998457324

Estudiante

Apellidos y Nombres: Toasa Supe María José

Cédula: 1804664819

Correo: maria.toasa4819@utc.edu.ec

Teléfono: 0992671039

Área de conocimiento: Información y comunicación (TIC)

Línea de investigación: Ciencias Informáticas para la modelación de Sistemas de Información a través del desarrollo de software.

Sublíneas de investigación de la carrera: Desarrollo de objetos de aprendizaje

2. INTRODUCCIÓN

Actualmente a nivel empresarial, en el sector privado como en el público se desarrolla software para automatizar o sistematizar procesos internos como externos de la empresa facilitando y agilizando los procesos supliendo así las necesidades. El proceso del desarrollo se ha enfocado en las tendencias tradicionales o monolíticos. Un sistema de Software se conforma íntegramente de programas que en su forma definitiva pueden ser ejecutables, además incluye manejos y definiciones de base de datos asociados a la configuración del software para su correcta ejecución, almacenando registros que servirán para realizar mantenimientos posteriores. Los mayores problemas se reflejan en el proceso del mantenimiento, porque por un nuevo cambio, una nueva petición o una corrección de errores que el cliente necesite significa el redespigie o caída de toda la aplicación porque todas las funcionalidades forman parte de un solo paquete generando una serie de fallos y errores, en cuanto a los aspectos de calidad, afecta la escalabilidad porque si en un futuro se requiere escalar un módulo representaría un problema en la empresa, de igual forma ocurre con la resiliencia debido a que al ocurrir un fallo por sobrecarga o caída en algún módulo de la aplicación se pierden todas las funcionalidades de la aplicación.

Las implementaciones monolíticas son un problema difícil de resolver, comprender y cambiar con el tiempo. Uno de los inconvenientes al momento de desarrollar un software son las limitaciones que se presentan durante la producción, dado que su elaboración empieza desde cero, además que, a diferencia de otras tecnologías, estas se rigen a extenuantes controles de calidad, sometándose a diversas combinaciones de condiciones, siendo este un factor clave en un mercado actualmente saturado por la competencia.

Como objetivo se va a implementar una aplicación basada en microservicios dado que la arquitectura monolítica siempre ha sido el enfoque tradicional, pero debido a las limitaciones de escalabilidad, la dificultad para mantener grandes bases de código, las actualizaciones de alto riesgo y los grandes costos iniciales, los desarrolladores de software se ven obligados a explorar diferentes enfoques que permitan el crecimiento de las aplicaciones sin representar la paralización o la caída total del sistema causando pérdidas a las empresas, además que el equipo de desarrollo puede implementar tecnologías adecuadas para cada microservicio de esta manera los futuros desarrolladores se enfocarán en entender un pequeño servicio y no toda la aplicación agilizando los cambios solicitados por los usuarios.

2.1. EL PROBLEMA

2.1.1. Situación Problemática

En la actualidad, en el mundo, las grandes empresas como Netflix, Google, Microsoft, IBM y DeepMind entre otras, buscan nuevas innovaciones en sus plataformas para que la empresa reaccione más rápido ante las nuevas demandas de los clientes y que el equipo de desarrollo trabaje de forma simultánea ofreciendo respuestas inmediatas, para cumplir estos objetivos han surgido la arquitectura de microservicios siendo un enfoque flexible a modo de dividir un monolito en pequeños servicios independientes de otros, así puedan ser desplegados sin que los fallos afecten a los demás.

El desarrollo de software en los países de América Latina se ha convertido en una de las herramientas más utilizadas para aumentar la productividad, ya que combina tecnología y provee soluciones para resolver diferentes problemas que se suscitan en una economía global, basada en la información y el conocimiento por lo cual un software desarrollado con la arquitectura de microservicios es más fácil de escalar y se puede implementar más rápido resolviendo así las necesidades de los clientes.

Las arquitecturas de software para el desarrollo de aplicaciones web de los sistemas de información en Ecuador han evolucionado a través del tiempo, desde los diseños de sistemas monolíticos primitivos, cliente-servidor y su dificultad de administración, así como las arquitecturas orientadas a microservicios, que describen procesos mediante el cual las aplicaciones se segmentan en una serie de partes o servicios más pequeños, en donde sus componentes se implementan para ser reutilizables con interfaces estándar para su invocación y que si se combinan permiten crear funcionalidades más complejas [1]. Por los aspectos antes mencionados se puede concluir que existe una gran necesidad de implementar una nueva arquitectura en el desarrollo de software con enfoque flexible que agilice y facilite el desarrollo de aplicaciones, permitiendo el mantenimiento de las aplicaciones sin necesidad de interrumpir las actividades de los usuarios.

2.1.2. Formulación del problema

¿El desarrollo de software se ve afectado por la arquitectura monolítica que carece de flexibilidad y facilidad para desarrollo de aplicaciones?

2.2. OBJETO Y CAMPO DE ACCIÓN

Objeto: Desarrollo de aplicaciones de software.

Campo: Aplicaciones basadas en microservicios.

2.3. BENEFICIARIOS

Tabla 1. Beneficiarios directos e indirectos

Beneficiarios Directos	Beneficiarios Indirectos
Grupos de Desarrollo de Software	Usuarios

2.4. JUSTIFICACIÓN

Un buen diseño de software debe estar orientado a mejoras futuras en el software, pero no siempre son tomadas en cuenta y los equipos de implementación sugieren de manera proactiva soluciones para satisfacer las nuevas necesidades requeridas. La investigación realizada contribuyó de manera óptima, generando nuevos conocimientos de las nuevas tendencias de arquitectura de software como son los microservicios, además de cómo las empresas y compañías están optando por trabajar con esta nueva arquitectura debido a que los microservicios trabajan mediante la colaboración de otros servicios más pequeños y autónomos. Por otro lado, el modelo de optimización propuesto en esta investigación ayuda a optimizar la dependencia y la complejidad en las aplicaciones basadas en arquitecturas de microservicios porque permite desplegar las aplicaciones sin perjudicar a los demás, así favoreciendo la escalabilidad, la facilidad de mantenimiento y aumenta la tolerancia a fallos.

Actualmente, existen diferentes limitaciones que se producen al momento de ejecutar aplicaciones que son desarrolladas con arquitecturas tradicionales o monolíticas, cuando se producen estos inconvenientes es necesario detener todos los procesos que se ejecutan en esta aplicación, por lo que la modificación o cambio no tiene nada que ver con los procesos que se realizan en ese momento. Otra de las limitaciones que se produce al ser aplicaciones monolíticas es la duplicación de código porque no comparten la funcionalidad de sus módulos, por lo que se requiere una única arquitectura, mantener o actualizar la funcionalidad de la aplicación es un

problema debido a su naturaleza monolítica. La arquitectura de microservicios fue planteada hace varios años debido a que los servicios se pueden agrupar para satisfacer las necesidades de uno o más clientes, desafortunadamente en ese momento no había tecnología para soportar esta arquitectura llamada microservicios.

2.5. HIPÓTESIS

Un modelo de optimización de dependencia y complejidad para el diseño de aplicaciones basadas en microservicios permitirá optimizar la dependencia y complejidad en los sistemas.

2.6. OBJETIVOS

2.6.1. General:

Determinar un modelo de optimización mediante la evaluación de métricas que identifican problemas de dependencia y complejidad, para el diseño de aplicaciones basadas en microservicios.

2.6.2. Específicos:

- Realizar una investigación sistemática sobre el objeto y campo de estudio para formular el marco teórico del proyecto.
- Analizar métricas de dependencia y complejidad para el diseño de aplicaciones basadas en microservicios.
- Aplicar el modelo más óptimo en la implementación de un caso práctico de microservicios.
- Validar la propuesta desarrollada, a través de la técnica de juicio de expertos.

2.7. SISTEMA DE TAREAS

Tabla 2. Actividades y tareas

OBJETIVO	ACTIVIDAD	RESULTADO DE LA ACTIVIDAD	DESCRIPCIÓN DE LA ACTIVIDAD
Objetivo Específico 1 Realizar investigación sistemática sobre el objeto y campo de estudio para formular el marco teórico del proyecto.	Buscar información de una variedad de fuentes y utilizar las teorías recopiladas en beneficio para el desarrollo del Marco Teórico	Selección de libros, artículos científicos y proyectos investigativos	Se utilizó para la investigación fuentes bibliográficas de libros y bases de datos
	Analizar la información recopilada que va a aportar a la investigación	Consolidar el marco teórico con información relevante	Se analizó la información recopilada para un mejor sustento de Fundamentación Científico Técnica
Objetivo Específico 2 Analizar métricas de dependencia y complejidad para el diseño de aplicaciones basadas en microservicios.	Investigar sobre las métricas de dependencia y complejidad en diferentes fuentes bibliográficas	Métricas de dependencia y complejidad	Análisis de las métricas para modelo de optimización de dependencia y complejidad
Objetivo Específico 3 Aplicar el modelo más óptimo en la implementación de un caso práctico de microservicios.	Desarrollar una aplicación basada en microservicios, utilizando en su diseño, el mejor modelo de optimización	Aplicación basada en microservicios	Se realizó el prototipo de la aplicación utilizando en su diseño, el mejor modelo de optimización
Objetivo Específico 4 Validar la propuesta desarrollada, a través de la técnica de juicio de expertos.	Presentar el trabajo propuesto a un grupo de expertos para que puedan determinar su validez	Se obtuvieron las críticas para la respectiva validación de la propuesta	Búsqueda de expertos Presentar la propuesta Aplicar el instrumento Análisis a través del coeficiente alfa de Cronbach

3. FUNDAMENTACIÓN TEÓRICA

3.1. ANTECEDENTES

En 2019 se realizó una propuesta de un sistema portátil para mejorar la gestión y seguimiento del aprendizaje, en la Escuela de Obstetricia Profesional de la Universidad Nacional de San Marcos. En esta propuesta han desarrollado un sistema que utiliza métodos flexibles con el objetivo de mejorar la gestión y control de su aprendizaje en su organización, y además de esta metodología, optan por utilizar la arquitectura de microservicios como herramienta para mejorar todos sus procesos. Al mismo tiempo que cubre diversas necesidades y carencias, como la mejora específica de la integración y el rendimiento, aprovechando al máximo las ventajas que ofrecen los microservicios y sus principales características, como la flexibilidad, la total independencia y la expresividad, gracias a la agregación de estas arquitecturas y metodologías flexibles. Refiriéndose al proyecto mencionado, queda claro que después de implementar un sistema, el objetivo principal pasa a ser mejorar los procesos que implementa a cualquier costo, para elegir la arquitectura de microservicios como una herramienta para crear un mejor rendimiento, transformándola en una eficiente, capaz de trabajar de la forma más adecuada y satisfacer así todos los requisitos evaluados.

En [2] utilizan la arquitectura basada en microservicios que es una nueva tendencia en la actualidad que se viene explotando en desarrollo de sistemas, tal enfoque es posible expandir con facilidad para asegurar la disponibilidad y mejorar el rendimiento. A partir de esta titulación, la tarea es presentar una propuesta de Ingeniería Microservicios basados en software de LCC OpenTech, C.A., en donde utilizará en el desarrollo del sistema de comercio electrónico. Con alta disponibilidad y escalabilidad.

Actualmente, el proceso de desarrollo de software que lleva a cabo la Coordinadora General de Tecnologías de la Información y las Comunicaciones (CGTIC) de la Asociación Nacional del Ecuador (ANE) constituye empleo de arquitectura de software tradicional o monolítica aprobada por el lenguaje de programación, antecedentes o experiencia del equipo de desarrollo utilizado; Debido al aspecto interrelacionado, este tipo de aplicaciones empaquetan toda la funcionalidad en una gran unidad de ejecución (un solo archivo o aplicación), lo que ha causado dificultades en varios aspectos como mantenimiento, escalabilidad y entregas como dice en [3].

3.2. APLICACIONES WEB

Son aplicaciones que contienen contenido y los usuarios pueden acceder mediante un servidor web a través del internet utilizando un navegador. Es decir, es una aplicación desarrollada en un lenguaje soportado por los navegadores web [4].

Una aplicación web es un tipo de software que utiliza un navegador como herramienta de interpretación y ejecución, y cabe señalar que el requisito básico para que se le denomine aplicación web es tener un servidor enlazado a través de Internet, pero el navegador debe también ser usado. Porque esta es la única forma de conectarse y jugar más tarde.

3.3. INGENIERÍA DE SOFTWARE

En [4] menciona que “la Ingeniería de Software es una disciplina de la ingeniería que se interesa por todos los aspectos de la producción de software”, con esto se puede comprender que se encarga de los métodos, procedimientos y posterior programación que requiere la producción de un Software, dado que su uso es muy variado es imperativo tomar en cuenta los requerimientos del usuario, asegurando que el producto final cumpla a cabalidad las expectativas previamente acordadas.

La Ingeniería en Software comprende diversas aplicaciones en el desarrollo de Software, sin embargo, para este proceso se requiere de múltiples recursos a diferencia de otras tecnologías, lo que dificulta su desarrollo por los controles de calidad a los que debe ser sometido. El diseño de Software es considerado como el pilar fundamental de los procedimientos de escala industrial, dado que se implantan de forma efectiva a diversos y muy delimitados procesos, garantizando de manera indirecta el éxito y sostenibilidad de la producción [4].

Esta tecnología se ocupa íntegramente del diseño y desarrollo de software, lo que a su vez se conoce como ciclo de vida del software, lo que permite llevar a cabo el proceso de desarrollo de software adaptado a cualquier aplicación, utilizando siempre el análisis, el diseño, la codificación y el despliegue. Conjunto de pasos es importante para el desarrollador cuando trabaja con métodos ágiles en el mundo del desarrollo de software.

3.3.1. Modelos de Proceso de Software

El Software atraviesa múltiples etapas, técnicas y procedimientos en su diseño, que serán cruciales al momento de determinar su ciclo de vida, resulta importante mencionar que el ciclo de vida se ve determinado por el marco en el que está desarrollado, dado que a través de esto

se garantiza un mínimo de garantía en los plazos de entrega y uso de recursos asignados, además de la garantía de calidad y las certificaciones necesarias para su funcionamiento como se observa en la Figura 1. [4]

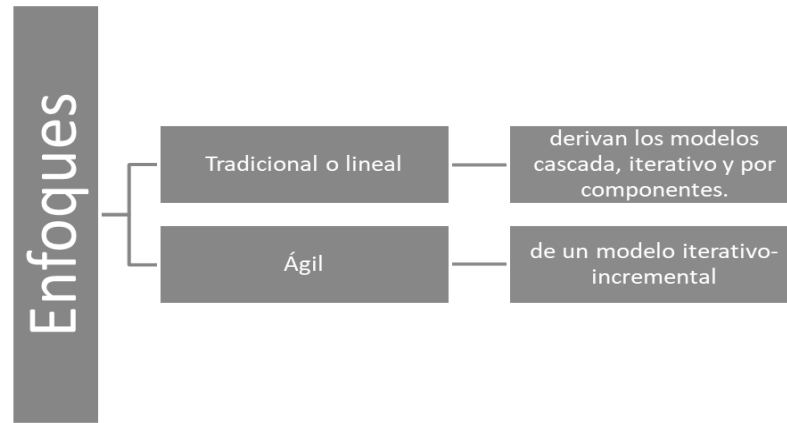


Figura 1.Tipos de enfoques según [5]

3.3.2. Modelo Clásico

En [5] menciona que el ciclo de vida más conocido es el modelo clásico que se subdivide en 6 etapas, como se puede observar en la Figura 2.

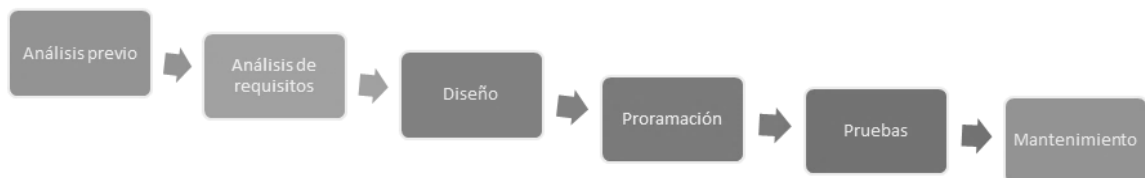


Figura 2. Etapas Modelo Clásico según [6]

También denominado ciclo de vida en cascada, en la primera etapa se realiza una especificación del Sistema, en la segunda etapa se realiza la especificación de los requisitos, en la tercera etapa la especificación del Diseño, en la cuarta etapa la especificación del Software, en la quinta etapa las Pruebas necesarias y en la etapa final el Mantenimiento, dado que en cada etapa se obtiene información necesaria para avanzar a la siguiente etapa, sin la opción de retorno hasta su terminación, sin embargo, al ser un procedimiento constituido de seis ciclos generan errores y datos inconclusos en sus primeras etapas, consumiendo mayor cantidad de recursos, siendo imposible generar requisitos completos y estables en su primer intento.

3.3.3. Modelos de proceso incremental

Este enfoque denominado iterativo e incremental, permite realizar de manera sectorizada un mismo proceso varias veces combinando los elementos de los procesos lineales y paralelos, como se puede observar en la figura 3.[6]



Figura 3. Flujo Procesos lineales según [7]

3.3.3.1. Prototipo

En [7] menciona al prototipo como “software provisional”, es utilizado para que los usuarios puedan confirmar lo que se muestra, lo que necesitan y se pueda preparar una nueva versión, teniendo en cuenta las modificaciones requeridas por el usuario, se realiza mediante la construcción de prototipos del Software, esto hace posible que la perspectiva se transforme poco a poco en un sistema real, algunos prototipos son construidos para un solo uso. El desarrollo puede iniciarse a lo largo del ciclo de vida en cascada desde una base más sólida, cuándo los prototipos pueden aclarar y validar los requisitos [7].

3.3.4. Modelo Espiral

En [5] menciona que es la combinación entre el modelo en cascada y prototipos, se desarrolla en una serie de bloques expandibles, el cual permite el desarrollo de versiones perfectas rápidamente. Tiene dos principales y distintivas características. El primero es un enfoque cíclico, aumentando gradualmente el determinismo y la implementación del sistema, mientras disminuye sus riesgos. El segundo es un conjunto de puntos de anclaje de referencia para asegurar el compromiso con soluciones posibles y satisfactorias para ambas partes [4].

3.3. METODOLOGÍAS TRADICIONALES

El enfoque tradicional impone disciplina de trabajo en el proceso de desarrollo de software, este enfatiza la planificación general de todo el trabajo que se necesita realizar y una vez hecho esto, según las condiciones climáticas, se iniciará el ciclo de desarrollo del producto de software [4].

Según [8] menciona que las metodologías tradicionales muchas de las veces son denominadas, de forma despectiva, es decir, como metodologías burocráticas. Porque centran toda su atención en llevar una documentación absoluta de todo el proyecto, la planificación y control, en especificaciones precisas del modelado y requisitos, además en cumplir con un plan de trabajo, definiendo esto, en la fase inicial del desarrollo del proyecto. Las metodologías tradicionales requieren de una disciplina estrictamente rigurosa de trabajo en el proceso de desarrollo de software, alcanzando así un software eficiente y de calidad. Por esta forma, hace énfasis en la planificación de todo el trabajo total a realizar y una vez que está todo detallado, empieza el ciclo de desarrollo del producto software. Entonces las metodologías tradicionales se centran específicamente en controlar los procesos mediante rigurosas definiciones de roles, artefactos, actividades, notaciones y herramientas para la documentación y el modelado bien detallado, por otro lado, las metodologías tradicionales no son adaptables a cambios futuros siendo esta una gran desventaja para trabajar con esta metodología porque los requisitos pueden variar o cambiar, además de sus altos costos en el momento de la implementación.

3.4. METODOLOGÍAS ÁGILES

Según [8] menciona que este enfoque ágil nace como respuesta a las dificultades y problemas que en algún momento puedan ocasionar las metodologías tradicionales, enfocándose en dos aspectos fundamentales, demorar la toma de decisiones y la planificación adaptativa. Basando su fundamento principal en la adaptabilidad de los procesos de desarrollo de software. Un modelo ágil de desarrollo, usualmente es un proceso incremental de entregas frecuentes con ciclos rápidos, también cooperativo con clientes y desarrolladores que trabajan en constante comunicación, sencillo por el procedimiento que es fácil de actualizar y aprender para el equipo, por último, es adaptativo porque es capaz de permitir cambios en el último momento. Las metodologías ágiles suministran una serie de pautas, condiciones y principios junto a técnicas adaptables que hacen que la entrega del proyecto sea menos complicada y más satisfactoria para los clientes y para el equipo de trabajo, así evitando los caminos tediosos de las metodologías tradicionales, de esta manera genera poca documentación y no hace el uso de métodos formales. Estas metodologías ágiles ponen de preferencia la capacidad de respuesta a un cambio que es más importante que el seguimiento estricto de un plan.

El desarrollo Ágil se refiere a la dirección de toma de decisiones de los proyectos de software y a los métodos de ingeniería de software enfocados en el desarrollo iterativo e incremental, en los que los requisitos y las soluciones evolucionan con el tiempo de acuerdo con las necesidades del proyecto. Las metodologías Ágiles se centran en las personas, valores, no en los procesos y planes, de esta manera reduce el riesgo demostrando valores y función durante el desarrollo, aumenta la productividad, sin aumentar la documentación y, si el proyecto no es factible, se puede corregir sus fallas rápidamente y sin consecuencias particulares. [4]

Tabla 3. Aspectos relevantes entre metodología tradicional vs metodología ágil

Metodología Tradicional	Metodología Ágil
Predictivos	Adaptativo, Flexible
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un solo proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Menos contacto con los clientes	Más contacto con los clientes, mayor participación del cliente
Distribuir el software al final del proceso de desarrollo.	Distribución constante del software
Documentación extensa	Se utiliza poca documentación

3.4.1. Scrum

Es un marco de trabajo utilizado en la gestión de proyectos de desarrollo de software, basada en la definición de roles, eventos, artefactos y reglas para compartir estos elementos. [9]

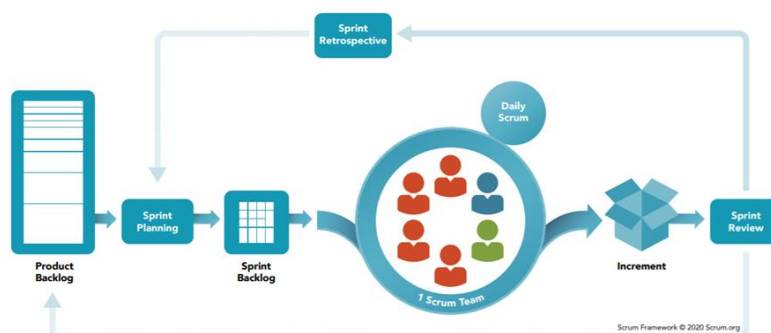


Figura 4. Flujo Procesos lineales según [7]

3.4.1.1. Roles principales

- **Product Owner:** Representa las partes interesadas internas y externas, por lo que debe comprender y apoyar las necesidades de todos los usuarios en el negocio, así como también las necesidades y el funcionamiento del Equipo Scrum. [4]
- **Scrum Master (o Facilitador):** Tiene como función que todo el equipo entienda Scrum y lo aplique de manera correcta, Su misión es que los equipos de trabajo alcancen sus objetivos hasta terminar la fase de sprint final, excluyendo todo tipo de dificultad que puedan presentarse en el camino. [4]
- **Scrum Team:** Son las personas responsables de asegurar que el equipo aporte valor al negocio. Representa las partes interesadas internas y externas, por lo que debe comprender y apoyar las necesidades de todos los usuarios en el negocio. [4]

3.4.1.2. Documentos

- **Product Backlog:** Para el Product Backlog se presenta un formato de las historias de usuario con la facilidad de detallar las necesidades de los usuarios del Sistema, aquí se definen los requisitos y prioridades asignándole un valor a cada historia de usuario con una estimación de días, añadiendo un día de holgura y se define en qué sprint será desarrollado cada Historia de Usuario.[4]
- **Sprint Backlog:** Conjunto de tareas en el que se observa en cuánto tiempo se realizará, con esto se revisa y se ajustan durante el curso del proyecto a intervalos regulares. [4]

3.4.1.3. Proceso

- **Historias de usuario:** Durante este proceso se generan historias de usuarios y criterios relacionados con las necesidades del propietario del producto, del cual se obtiene información relevante para obtener de una manera rápida los requisitos. [4]
- **Requerimientos de historia de usuario:** Durante este proceso se reemplazan documentos por conversaciones que permiten obtener cual es la necesidad, la prioridad, tamaño y el diseño de cada historia de usuario.[4]
- **Creación de tareas:** Durante este proceso las historias de usuario son revisadas y aprobadas por los usuarios, posteriormente se dividen en actividades específicas y se combinan en listas de actividades. [4]
- **Estimación de tareas:** Durante este proceso el equipo Scrum estima el esfuerzo

requerido para completar cada actividad en la lista, con esto se realiza una planificación para completar cada actividad. [4]

- **Creación sprint Backlog:** Para realizar este proceso el equipo Scrum tiene una reunión que tiene lugar al comienzo de cada Sprint, a medida que se determinan las tareas a crear, se crea un sprint backlog rápido que incluye todas las tareas a realizar en el sprint.[4]

3.4.2. Método de MosCoW

La metodología MoSCoW se basa en el hecho de que si bien todos los requisitos son importantes, es necesario resaltar aquellos que aportan el mayor valor al sistema, que permite enfocar los trabajos de desarrollo de manera más eficiente.[10]

La técnica MoScoW inventada por Dai Clegg, permite determinar la priorización en qué se debe hacer primero, qué se debe hacer después y qué no se necesita hacer, obteniendo así la definición de prioridades Alta, Media, Baja asignadas a cada Historia de Usuario.[10]

3.4.3. Release planning

Release planning se refiere al uso de ciclos cortos de desarrollo (iteraciones) que le muestran software terminado al cliente y obtienen retroalimentación de él. La definición de terminado está relacionada con las pruebas de aceptación.[11]

3.4.4. Planinig Poker

Se trata de una técnica de apreciación en la que participan diferentes miembros del grupo. El proyecto (o al menos una parte suficientemente representativa del mismo) y el proceso Llegar a un consenso sobre el tiempo requerido para completar la tarea[12].

3.4.5. DevOps

DevOps es una metodología que permite cambiar la forma por la cual se administra y se lleva a cabo el ciclo de desarrollo de software a nivel técnico. El equipo de trabajo y desarrollo descartan el trabajo por partes para trabajar en una colaboración de dos caminos. Estos equipos cubren el ciclo de vida completo y todo el desarrollo de software, así garantizando procesos más seguros, rápidos, y tener una entrega del alta calidad y productos más confiables.[13]

Según [14] es un enfoque ágil con el objetivo de reducir los tiempos necesarios dentro del ciclo de vida software como la migración de versiones, ejecución de pruebas, detección de bugs, etc.

Según [13] al incluir DevOps en el proyecto nos tenemos que referir a la integración continua, que es una práctica que permite automatizar tareas cuando se produce un evento. Entre las acciones y procesos que se pueden automatizar están las siguientes: [13]

- Compilación de componentes
- Pruebas unitarias
- Pruebas de integración
- Calidad de código
- Escalado
- Despliegue de software

3.4.5.1. Áreas de DevOps

- En 2012, Debois diseñó áreas claves para indicar los aspectos fundamentales en DevOps. En la Figura 5 se muestra la interacción entre el desarrollo y las operaciones es bidireccional, por qué incide en las tareas que fomentan el intercambio del conocimiento y la estimación entre los equipos de trabajo.[15]
- **Área 1:** Extender la entrega a producción: Los equipos de desarrollo y operaciones colaboran para mejorar el proceso de entrega de un proyecto al entorno de producción. [15]
- **Área 2:** Extender la respuesta del sistema en operaciones al proyecto: Hacer extensible toda la información relevante en producción al equipo de desarrollo. [15]
- **Área 3:** Compartir todo el conocimiento del proyecto al equipo de operaciones: El equipo de desarrollo comparte la responsabilidad de todo lo que ocurre en el entorno de producción.[15]
- **Área 4:** Integrar el conocimiento de operaciones en el equipo de desarrollo: Operaciones debe involucrarse en el proyecto desde el inicio.[15]

3.5. CALIDAD DE SOFTWARE

Según [9] menciona que, el concepto de calidad del software es "un acuerdo de desempeño y requisitos funcionales claramente establecidos con estándares de desarrollo completamente documentados y características inherentes que se esperan de todo software desarrollado por expertos", con los requisitos funcionales y no funcionales detallados en la etapa de análisis, es

importante atender esta demanda con las características mínimos, estándares y estándares de calidad requeridos en cada etapa del ciclo de vida del programa de manera que se minimice el riesgo de falla del proyecto [16]. Por otro lado, IEEE establece que la calidad es el grado en que un sistema, proceso o componente cumple con los requisitos, necesidades o expectativas de los usuarios [16].

3.5.1. Clasificación de la calidad del software

La calidad de software se clasifica de acuerdo al nivel de proceso, al producto o a la calidad de uso dependiendo del enfoque de evaluación. [17]

Según [17] manifiesta que “Entiéndase los atributos de calidad de un sistema software como las propiedades o cualidades que la aplicación debe satisfacer. Existe una categorización de dichos atributos en el standard ISO/IEC FCD 25010: Systems and software engineering-and software quality models, que agrupa en las siguientes categorías principales: funcionalidad, compatibilidad, fiabilidad, usabilidad, eficiencia, seguridad, mantenibilidad y portabilidad”.

Los atributos de calidad de software en microservicios se basan en los estándares de la IEEE Std. 1061, ISO/IEC 25010:2011 y MITRE Guide to Total Software Quality Control que puntualiza las propiedades de calidad de un software. En la Tabla 3.4 se muestran los estándares y atributos de calidad definidos. [17]

Tabla 4. Estándares y atributos de calidad

IEE Std. 1061	ISO/IEC 25010:2011	MITRE Guide to Total Software Quality Control
<ul style="list-style-type: none"> - Eficiencia - Funcionalidad - Mantenibilidad - Portabilidad - Confiabilidad - Usabilidad 	<ul style="list-style-type: none"> - Eficiencia de desempeño - Adecuación Funcional - Mantenibilidad - Portabilidad - Confiabilidad - Usabilidad - Compatibilidad - Seguridad 	<ul style="list-style-type: none"> - Eficiencia - Integridad - Confiabilidad - Supervivencia - Usabilidad - Correctitud - Mantenibilidad - Testabilidad - Expansibilidad - Flexibilidad - Interoperabilidad - Portabilidad - Reusabilidad

3.5.2. Norma ISO 9126

ISO 9126 es un estándar internacional para la evaluación del Software [17]. Está supervisado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos [17]. El estándar está dividido en cuatro partes, las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso. [17]

El modelo de calidad se estableció en la primera parte de la norma ISO 9126-1. Dicho estándar ha sido desarrollado en un intento de identificar los atributos clave de calidad para el software. [17]

3.5.2.1. Evolución de la norma ISO 9126

ISO/IEC 9126 (1991): Evaluación de productos de software. Se desarrollaron características de calidad y pautas para su uso para satisfacer estas necesidades, identificando seis características de calidad y describiendo un modelo de proceso de calidad y evaluación del producto software.[9] ISO/IEC TR 9126-3:2003 proporciona métricas internas para medir atributos de seis características de calidad externas definidas en ISO/IEC 9126-1. ISO/IEC TR 9126-2 define métricas externas, ISO/IEC TR 9126-3:2003 define métricas internas e ISO/IEC 9126-4 define métricas de calidad en uso, para la medición de las características y subcaracterísticas [18].

Las métricas internas miden el software en sí, las métricas externas miden el comportamiento del sistema informático que incluye el software y las métricas de calidad en uso miden los efectos del uso del software en un contexto de uso específico.[18]

3.5.2.2. Factores de la calidad ISO 9126

La norma ISO 9126 se desarrolló para definir las principales características de los programas informáticos. Este sistema define seis atributos básicos de calidad:

- **Funcionalidad.** Grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad. [18]
- **Confiabilidad.** Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes atributos: madurez, tolerancia a fallas y recuperación. [18]
- **Usabilidad.** Grado en el que el software es fácil de usar, según lo indican los siguientes subatributos: entendible, aprendible y operable. [18]
- **Eficiencia.** Grado en el que el software emplea óptimamente los recursos del sistema,

según lo indican los subatributos siguientes: comportamiento del tiempo y de los recursos. [18]

- **Facilidad de recibir mantenimiento.** Facilidad con la que pueden efectuarse reparaciones al software, según lo indican los atributos que siguen: analizable, cambiable, estable, susceptible de someterse a pruebas. [18]
- **Portabilidad.** Facilidad con la que el software puede llevarse de un ambiente a otro según lo indican los siguientes atributos: adaptable, instalable, conformidad y sustituible. [18]

Igual que otros factores de la calidad del software estudiados en las subsecciones anteriores, los factores ISO 9126 no necesariamente conducen a una medición directa [18]. Sin embargo, proporcionan una base útil para hacer mediciones indirectas y una lista de comprobación excelente para evaluar la calidad del sistema. [18]

3.5.3. Calidad en microservicios

Según [6] menciona que “Con los microservicios, hemos añadido otro nivel de complejidad, entender los diferentes tipos de prueba que podemos ejecutar es importante para ayudarnos a equilibrar fuerzas o a veces opuestas, de poner que nuestro software tiene la calidad suficiente.”

3.5.3.1. Atributos de calidad de microservicios

Los atributos de calidad son esenciales para las aplicaciones actuales [19]. La disponibilidad, el rendimiento, la escalabilidad, seguridad, mantenibilidad, y la tolerancia a fallos son características esenciales que toda aplicación debe manejar [19]. Una arquitectura basada en microservicios permite una gestión independiente de los atributos de calidad, según la necesidad específica de cada microservicio.[19] El tamaño y el número de microservicios que componen una aplicación afectan directamente a sus atributos de calidad. [19]

Otras métricas que están directamente relacionadas con los atributos de calidad son el costo de despliegue, el costo de aseguramiento de la calidad, el impacto en la seguridad y el impacto de la escalabilidad. [19]

3.5.3.2. Problemas de calidad de microservicios

Según [20] los problemas o inconvenientes presentados en microservicios son:

- **Complejidad de gestión.** - Si el número de servicios aumenta, mayor número de nodos

que mantener, por lo que la integración, las pruebas y la gestión del producto completo es más complicado [20]. Por lo tanto, se requiere de una infraestructura para el independiente escalado de los servicios, que soporten tolerancia o fallos y continúan entrega. Esto puede incluir el uso de virtualización. [20]

- **Coordinación/Integración.** – Numerosos servicios que se desarrollan y operan de forma independiente deben ser integrados [20]. Aunque cada micro servicio pueda ser simple, la integración es mucho más complicada, por las dependencias y llamadas que existen entre ellos [20]. Por otra parte, y con el fin de ofrecer un servicio común, deben definirse mecanismos de coordinación y control de las distintas versiones. [20]
- **Dependencia/Comunicación entre servicios.** – Las dependencias entre servicios requieren de una adecuada infraestructura de comunicaciones, ya que las llamadas a funciones locales (en otro tipo de sistemas), se convierten en llamadas remotas a funciones locales (en otro tipo de sistemas), se convierten en llamadas remotas a otros servicios lo que afecta al rendimiento del sistema[20].

3.5.3.3. Cohesión y acoplamiento

Según [6] la Alta cohesión pretende que “Los comportamientos relacionados se sitúen juntos y que no lo relacionado se sitúen en otro lugar. ¿Por qué? Bueno, si queremos cambiar el comportamiento, queremos ser capaces de cambiarlo en un lugar, y liberar ese cambio tan pronto como sea posible. Si tenemos que cambiar ese comportamiento en muchos lugares diferentes, tendremos que lanzar muchos servicios diferentes, quizás al mismo tiempo para realizar ese cambio.”

Cuando los servicios están débilmente acoplados, un cambio en un servicio debería requerir un cambio en otro. El objetivo de un micro servicio es poder hacer un cambio en un servicio y desplegarlo, sin necesidad de cambiar ninguna otra parte del sistema. [6]

3.5.3.4. Dependencia

Según [20] menciona que la “Dependencia o Comunicación entre servicios son dependencias entre servicios que necesitan una infraestructura de comunicación adecuada, ya que las llamadas funciones locales (en otro tipo de sistemas), se convierten en llamadas remotas a otros servicios, afectando el rendimiento del sistema”.

La dependencia se define al recuento de otros servicios que dependen de este servicio, esta métrica es directamente proporcional a la autosuficiencia del microservicio dado que debe tener su propia autonomía.[19]

3.5.3.5. Complejidad

Dividir nuestro sistema en micro servicios más pequeños y detallados da como resultado múltiples beneficios. Sin embargo, también agrega complejidad cuando se trata de monitorear el sistema en producción [6].

La complejidad de los microservicios debe ser baja, para que puedan ser modificados en un par de semanas, reescritos y mejorados rápidamente. Si la complejidad es alta, entonces el costo del cambio es mayor. La medición de la complejidad es fundamental para desarrollar aplicaciones basadas en microservicios.[21]

Por otro lado, la complejidad en el interior de un componente de microservicios y las conexiones entre todos los componentes lleva a un lugar más explícito y más complicado de inspeccionar. Es fácil suponer que las cosas mejorarán cuando se mira al interior de los componentes, perdiendo así las conexiones entre los servicios. [22]

3.5.3.6. Como la dependencia y complejidad afecta en la calidad de las aplicaciones basadas en microservicios.

La creación de más microservicios puede afectar a la mantenibilidad porque los costes de las pruebas aumentarán, más aún si no se dispone de pruebas automatizadas [21]. Por otra parte, el rendimiento también puede verse afectado por tener que integrar y procesar datos de varias aplicaciones distribuidas [21]. Es evidente que los atributos de calidad se ven afectados por la granularidad de los microservicios, por lo que deben tenerse en cuenta al definir un modelo, método o técnica para determinar la granularidad.[21]

La dependencia y la complejidad en los micro servicios afectan a los atributos de calidad como a la flexibilidad y a la mantenibilidad porque existen una serie de desventajas en un sistema basado en micro servicios, el aumento de complejidad para gestionar o realizar un test de las dependencias, es un sistema no uniforme porque conlleva diseños y arquitectura no semejantes de esta manera aumenta el coste de mantenimiento a corto y a largo plazo.[22]

Cada servicio contiene su propia lógica de negocio y se despliega por separado [13]. Esto hace posible cambiar y extender gradualmente con nuevas características sin afectar al resto del sistema. La propiedad autónoma permite la flexibilidad en la selección de la tecnología más adecuada para un servicio específico. [13]

3.6. ARQUITECTURAS DE SOFTWARE

Según [6] menciona que la arquitectura de Software se refiere a la estructura de un sistema, describe su forma, sus componentes y la manera en la que se adaptan entre ellos.

3.6.1. Arquitectura de software monolítica

En [9] menciona que la arquitectura monolítica consiste en crear una aplicación autónoma que contenga absolutamente todas las funciones necesarias para realizar la tarea para la que fue diseñada, sin dependencias externas adicionales. En este sentido, sus componentes trabajan juntos, compartiendo los mismos recursos y la misma memoria. En pocas palabras, una aplicación monolítica es una unidad de código vinculado.

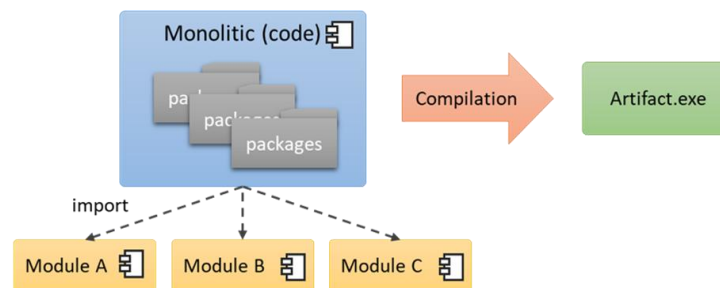


Figura 5. Arquitectura Monolítica según [10]

Según [9] menciona que un software utilizando arquitectura monolítica se puede crear como una sola unidad de software o constar de varios módulos o bibliotecas, pero lo que lo hace diferente es que cuando se compila lo hace como una parte, por lo que todos los módulos y bibliotecas se agregaron con la aplicación principal.

3.6.2. Arquitecturas de software basadas Microservicios

Este es un enfoque para desarrollar una sola aplicación como un conjunto de microservicios, cada uno operando de acuerdo con su propio proceso y un mecanismo de comunicación liviano, a menudo un recurso de una interfaz de programación de aplicaciones (API) sobre protocolo de transferencia de hipertexto (HTTP) [11]. Estos servicios están contruidos alrededor de las capacidades del negocio y con independencia de despliegue e implementación totalmente automatizada [11]. Existe un mínimo de gestión centralizada de estos servicios, los que pueden estar escritos en lenguajes de programación diferentes y utilizar diferentes tecnologías de almacenamiento de datos [11].

El término microservicios no es relativamente nuevo, este estilo arquitectónico fue acuñado por Martin Fowler durante una Conferencia de Arquitectos de Software como una descripción del nuevo campo. No existe una definición en concreto para microservicio, sin embargo, una aproximación que la realiza [7] lo define como: “Pequeños servicios autónomos que trabajan juntos”. Una arquitectura de microservicios promueve el desarrollo y despliegue de aplicaciones compuestas por unidades independientes, autónomas, modulares y auto-contenidas, lo cual difiere de la forma tradicional o monolítica [11].

3.6.2.1. Características de la arquitectura basada en microservicios

Para [23] las arquitecturas basadas en microservicios cuentan con las siguientes características:

- **Componentización a través de servicios.** Una razón principal para usar los servicios como componentes es que los servicios se pueden implementar de forma independiente [23]. De esta forma, al realizar un cambio en cualquier componente, no hará falta volver a implementar toda la aplicación, sino volver a desplegar el componente modificado [23]. Sin embargo, algunos cambios cambiarán las interfaces del servicio y, como consecuencia, una cierta coordinación, pero el objetivo de una buena arquitectura de microservicio es minimizarlos mediante límites cohesivos de servicio y mecanismos de evolución en los contratos de servicio [23].
- **Organizado en torno a las capacidades empresariales.** El enfoque de microservicio para la división es diferente, dividiéndose en servicios organizados en torno a la capacidad empresarial [23]. Debido a que dichos servicios requieren una implementación amplia de software para esa área de negocios, incluida la interfaz de usuario, el almacenamiento persistente y cualquier colaboración externa, los equipos son multifuncionales, incluyendo la gama completa de habilidades requeridas para el desarrollo: experiencia del usuario, base de datos y gestión de proyectos. [23]
- **Productos, no proyectos.** La mentalidad del producto, se enlaza con la vinculación con las capacidades empresariales[23]. En lugar de considerar el software como un conjunto de funcionalidades para completar, existe una relación continua en la que la pregunta es cómo puede el software ayudar a sus usuarios a mejorar la capacidad empresarial. [23]
- **Gestión descentralizada de datos.** La gestión descentralizada de datos se presenta de diferentes maneras[23]. En el nivel más abstracto, significa que el modelo conceptual del mundo será distinto en cada uno de los servicios[23]. Además de descentralizar las decisiones sobre los modelos conceptuales, los microservicios también descentralizan

las decisiones de almacenamiento de datos, puesto que prefieren dejar que cada servicio administre su propia base de datos, ya sea en diferentes instancias de la misma tecnología de base de datos, o en sistemas de bases de datos completamente diferentes. [23]

- **Diseño para fallas.** Las aplicaciones basadas en microservicios deben diseñarse de modo que puedan tolerar la falla de cualquiera de los servicios que la componen [23]. Una llamada a un servicio puede fallar en cualquier momento debido a la indisponibilidad del proveedor, y el cliente debe responder a esta de la mejor manera posible[23]. Por esta razón, es importante poder detectar las fallas rápidamente y, si es posible, restaurar el servicio automáticamente[23]. Las aplicaciones de microservicio ponen mucho énfasis en el monitoreo en tiempo real de la aplicación, verificando elementos como cuántas solicitudes por segundo está obteniendo, la base de datos y la cantidad de pedidos por minuto que se reciben [23]. Esta actividad es vital para detectar el mal comportamiento emergente rápidamente, de modo que pueda solucionarse. [23]
- **Diseño evolutivo.** Cuando se intenta dividir un sistema de software en componentes, se enfrenta a la decisión de cómo dividir las piezas [23]. La propiedad clave de un componente es la noción de reemplazo y la capacidad de actualización de forma independiente, lo que implica que se deben buscar puntos en los que se podría imaginar reescribir un componente sin afectar a sus colaboradores[23].

3.7. HERRAMIENTAS DE DESARROLLO

3.7.1. Backend

3.7.1.1. Eureka

Eureka es una herramienta de descubrimiento que permite buscar simultáneamente en multitud de recursos, tanto en formato impreso como electrónico, mediante una única consulta y de forma muy simple.[24]

Con todo lo anterior, se puede entender que Eureka es un servidor que debe usarse si trabaja con servicios y servicios precisos, especialmente porque se puede guardar y ubicar sin problemas. El tema, porque este servidor tiene la capacidad de formar energía para comunicarse y comunicarse entre pequeños servicios a través de grupos de procesos básicos a través de los cuales todo tipo de información habla con control y verificación, y esto hace que este servidor y el uso de este pequeño. El servicio es exactamente tolerancia con el accidente, esto tolera que es posible identificar y resolver este problema. Es necesario mencionar un hecho importante

para resaltarlo, saber que este servidor debe ser desarrollado por Netflix, es necesario mencionar que esta gran empresa tiene la introducción.

3.7.1.2. Framework Spring Cloud

Spring Cloud ofrece un modelo de programación en un entorno Java, que está construido sobre Spring Boot[24]. Este framework, tiene una relación directa con el lenguaje de programación Java y lo utiliza como entorno virtual, facilita y proporciona puertos y configuraciones que facilitan la gestión y el desarrollo de servicios. El micro se convierte en una tarea sencilla de realizar, debido a los atributos de calidad utilizados en estos y modelos, que son las características básicas de esta arquitectura, gracias a todas las ventajas que ofrece este marco es mejorar el rendimiento de las aplicaciones construidas en su plataforma, mientras que esto también reduce significativamente el tiempo de desarrollo y, por lo tanto, reduce los costos, en términos de estimación de costos, tiempo y esfuerzo.

3.7.1.3. Base de datos

Una base de datos es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos [25]. 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 [25].

La base de datos en sí es una información organizada y bien estructurada almacenada en el sistema, básicamente permite el almacenamiento de datos, la búsqueda y el ingreso de nuevos datos, y todas estas operaciones se realizan de manera sencilla y eficiente desde la base de datos. La computadora y los datos ingresados son almacenados, registrados y organizados a través de tablas preestructuradas por defecto.

3.7.1.4. PostgreSQL

PostgreSQL es un gestor de bases de datos orientado a objetos (SGBDOO u ORDBMS por sus siglas en inglés) muy conocido y utilizado en entornos de software libre por ser responsive de 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 [25].

PostgreSQL tiene muchos pros y contras, pero si sabe cómo aprovecharlo, puede usarse principalmente para almacenar y procesar una cantidad de datos e información (DWH) sin precedentes, y también puede usarse en servicios web para todos los propósitos. En una

aplicación existe un historial y evidencia de muchas plataformas que usan este administrador de base de datos como fuente de servicio, para asegurarnos de que tenemos un caso de que Skype lo está usando como un servicio para sus diversos servicios de telefonía y más, una herramienta de código abierto que significa una gran contribución para aquellos que eligen y pueden usarlo.

3.7.1.5. MariaDB

Hoy en día, MariaDB y PostgreSQL son muy asequibles y se encuentran entre los sistemas mejor calificados en el ranking mundial, ocupando las posiciones 2 y 5 respectivamente. Ambos son sistemas de código abierto, por lo que la atención se centra en sistemas asequibles con buenas características que muestren su solidez y altos niveles de rendimiento. Las etiquetas también muestran que están siendo utilizadas por una gran cantidad de administradores que buscan un DBMS totalmente funcional para las aplicaciones que utilizan para almacenar datos relacionales y basados en SQL [25].

3.7.1.6. 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 [25].

MongoDB, a diferencia de otras bases de datos, en lugar de almacenar datos e información en registros mediante tablas, almacena estos datos en tipos de estructuras que pueden llamarse documentos, gracias a las cuales se guardan archivos. Bson, que es una representación de una estructura Json, pero convertida a binaria, es una base de datos no relacional sin relaciones entre relaciones o entre claves, lo que la convierte en una gran base de datos. La base de datos es más flexible que SQL o la base de datos relacional, y dado que MongoDB, que no es relacional, no es compatible con SQL, por lo que no tiene una lista de materiales escrita en sí misma, lo que significa que no admite relaciones o claves externas, ni tablas sino conjuntos de documentos, como se indicó anteriormente.

3.7.1.7. Spring Tools IDE(Eclipse)

Spring es un framework de desarrollo y contenedor de inversión de control open source para crear aplicaciones Java[26].

Sprint tiene como una de sus características más notables la inyección de dependencias. La inyección de dependencias permite instanciar objetos sin que tengamos que crear instancias dentro del código como `new`. Esto facilita en gran medida reducir el acoplamiento entre clases, haciendo que estas sean más sencillas (POJO) y también facilita las pruebas unitarias[26].

3.7.1.8. Maven

Maven es una herramienta de gestión de proyectos de software de código abierto Por gestión nos referimos a gestionar el ciclo de vida desde la creación de un proyecto en un lenguaje específico hasta la creación de un binario que se puede distribuir con el proyecto [27].

3.7.2. Frontend

3.7.2.1. JavaScript

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos [28]. En otras palabras, los programas escritos en JavaScript se pueden probar directamente en cualquier navegador sin necesidad de operaciones intermedias [28].

Para entenderlo mejor, JavaScript es un lenguaje de programación como muchos otros lenguajes, pero con características que lo hacen único, es un lenguaje especial y diferente porque es un lenguaje de comandos que te permite expresar tus ideas directamente usando una aplicación web, actualmente es el más utilizado en el mundo de la programación, ya que es un lenguaje explicado que no necesita compilar, HTML y CSS son técnicas vinculantes. Juntarlos es un complemento perfecto para este lenguaje, con esta combinación se puede construir cualquier tipo de aplicación web sin inconvenientes ni limitaciones.

3.7.2.2. 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 [29].

Básicamente, Angular es un framework desarrollado basándonos en el código JavaScript original, con este framework puedes desarrollar fácilmente aplicaciones web en una página, y una de sus ventajas es si necesitas actualizar parte de la aplicación se está construyendo, puede

hacerlo y evitar actualizar toda la aplicación como sucede con otros marcos y lenguajes de programación, lo que puede convertirse en un gran problema en algún momento, pero con angular es un código reactivo y de secuencias de comandos que cuando se realizan cambios, se procesa automáticamente sin necesidad de actualizar toda la página.

3.8. CÁLCULO DE LA COMPLEJIDAD Y LA DEPENDENCIA EN LA ARQUITECTURA BASADA EN MICROSERVICIOS

3.8.1. Fórmulas para el cálculo de la dependencia

3.8.1.1. Propuesta 1

El acoplamiento mide el grado de dependencia de un componente de software en relación con otro. En este caso el acoplamiento se definió por tres métricas: 1) importancia absoluta del microservicio (AIS), 2) dependencia absoluta del microservicio (ADS), y 3) interdependencia de los microservicios (SIY). Estas métricas se calculan con base a las dependencias de las historias de usuario para cada microservicio [21].

Importancia absoluta del microservicio (AIS): AIS es el número de otros microservicios Invoque al menos un proceso para la interfaz de microservicio. AIS_i es el número de clientes (otros microservicios) que invocan al menos una operación de ms_i . A nivel de sistema, se define el vector \vec{AIS} , que contiene el valor AIS calculado para cada microservicio.

Para calcular el valor total de AIS a nivel de sistema ($AisT$), se calcula la norma del vector \vec{AIS} . Donde n es el número de microservicios de MSBA, por lo tanto[21]:

$$\vec{AIS} = [AIS_1, AIS_2, AIS_n] \quad (1)$$

Ecuación 1. Importancia absoluta del microservicio

$$AisT = \left| \vec{AIS} \right| = \sqrt{AIS_1^2 + AIS_2^2 + \dots + AIS_n^2} \quad (2)$$

Ecuación 2. Importancia absoluta total del microservicio

Dependencia absoluta del microservicio (ADS): ADS es la cantidad de otros microservicios de los que depende el microservicio ADS_i es una pequeña cantidad de servicios mediante los

cuales se llama al menos una operación de ms_i . Para calcular el valor total de ADS a nivel de sistema ($AdsT$), se calcula la norma del vector ADS . Entonces[21]:

$$\vec{\rightarrow}_{ADS} = [ADS_1, ADS_2, \dots, ADS_n] \quad (3)$$

Ecuación 3. Dependencia absoluta del microservicio

$$AdsT = \left| \vec{\rightarrow}_{ADS} \right| = \sqrt{ADS_1^2 + ADS_2^2 + \dots + ADS_n^2} \quad (4)$$

Ecuación 4. Dependencia absoluta total del microservicio

Interdependencia de los microservicios (SIY): El número de pares de microservicios que dependen unos de otros. En este caso, SIY_i define el número de pares de microservicios que dependen bidireccionalmente uno del otro dividido por el número total de microservicios. A nivel de sistema, se definió el vector SIY [21]:

$$\vec{\rightarrow}_{SIY} = [SIY_1, SIY_2, SIY_n] \quad (5)$$

Ecuación 5. Interdependencia de los microservicios

$$SiyT = \left| \vec{\rightarrow}_{SIY} \right| = \sqrt{SIY_1^2 + SIY_2^2 + \dots + SIY_n^2} \quad (6)$$

Ecuación 6. Interdependencia total de los microservicios

Se definió el vector $\vec{\rightarrow}_{Cp}$ como el vector que contiene las métricas de acoplamiento calculadas para MSBA, calculando la norma del vector $\vec{\rightarrow}_{Cp}$ tenemos el valor de acoplamiento para la aplicación (CpT) [21]:

$$\vec{\rightarrow}_{Cp} = [AisT, AdsT, SiyT] \quad (7)$$

Ecuación 7. Acoplamiento de los microservicios

$$CpT = \left| \vec{\rightarrow}_{Cp} \right| = \sqrt{AisT^2, AdsT^2 + SiyT^2} \quad (8)$$

Ecuación 8. Acoplamiento total de los microservicios

Cohesión de MSBA

La cohesión y el acoplamiento son dos propiedades contrastantes[21]. Una solución que equilibre la alta cohesión y el bajo acoplamiento es el objetivo de los desarrolladores cuando implementan aplicaciones basadas en microservicios. Utilizamos las siguientes métricas de cohesión[21]:

- **Falta de cohesión (LC):** LC medir el número de pares de microservicios que no dependen unos de otros, LC de ms_i fue definida como el número de pares de microservicios que no tienen ninguna interdependencia con ms_i [21].
- **Grado de la falta de cohesión (Coh):** El grado de cohesión Coh de cada microservicio se define como la proporción de la métrica de falta de cohesión dividida por el número total de microservicios que forman parte de la aplicación; de la siguiente forma [21]:

$$Coh_i = Lc_i/n \quad (9)$$

Ecuación 9. Falta de cohesión

Donde n es el número de microservicios. A nivel de sistema, se definió el vector \vec{Coh} , calculando la norma del vector \vec{Coh} tenemos el valor de cohesión para MSBA ($CohT$) [21]:

$$\vec{Coh} = [Coh_1, Coh_2, \dots, Coh_n] \quad (10)$$

Ecuación 10. Grado de falta de cohesión

$$CohT = \left| \vec{Coh} \right| = \sqrt{Coh_1^2 + Coh_2^2 + \dots + Coh_n^2} \quad (11)$$

Ecuación 11. Grado total de falta de cohesión

3.8.1.2. Propuesta 2

El acoplamiento de un servicio se describe como la fuerza de la dependencia entre servicios en el sistema, es evaluado como el promedio del número de servicios conectados directamente y está dado por [30]:

$$acoplamiento = \frac{\text{número de servicios conectado}}{\text{número de servicios}} = \frac{n_c + n_p}{n_s} \quad (12)$$

Ecuación 12. Acoplamiento

Donde:

n_c = Número de servicios consumidores

n_p = Número de servicios pendientes

n_s = Número total de servicios

La cohesión se describe la fuerza de la relación entre las operaciones en un servicio, es evaluado como el inverso del promedio del número de mensajes usados, y está dada por [30]:

$$\text{cohesión} = \frac{n_s}{M(S)} \quad (13)$$

Ecuación 13. Cohesión

Donde:

$M(S)$ = Es el conjunto de mensajes de las operaciones de un servicio

n_s = Número total de servicios

3.8.1.3. Propuesta 3

Cuando se dice que MS1 está acoplado a MS2, se quiere decir que MS1 depende de MS2 para completar una cierta parte de su propia funcionalidad. Sin embargo, no significa necesariamente lo mismo, en modo inverso, se puede decir que MS1 está acoplado a MS2 cuando MS1 inicia una comunicación de solicitud/respuesta con MS2. Una buena analogía es imaginar una arquitectura cliente-servidor donde MS1 es el cliente y MS2 es el servidor, el cliente depende del servidor, no al revés. En este sentido, se entiende que existe un acoplamiento unilateral o bilateral. Una relación entre dos microservicios puede ser jerárquico (exclusivamente un servicio utiliza los servicios de otro), o bidireccional (ambos servicios utilizan servicios del otro servicio)[31].

Para medir qué tan acoplado está MS1 a MS2, necesario contar, para cada proceso funcional de MS1, el número de movimientos de datos que son intercambiados entre MS1 y MS2, para todos los casos en los que MS1 inicia la comunicación con MS2[31].

$$\text{PF1: } (eXit) + (\text{entry}) = \text{CFP} \quad (14)$$

Ecuación 14. Llamadas y solicitudes

Donde:

(eXit): Número de llamadas del microservicio.

(entry): Número de solicitudes del microservicio.

$$AcpT = ms_1 + ms_2 + \dots + ms_n \quad (15)$$

Ecuación 15. Acoplamiento total del microservicio

3.8.2. Fórmulas para calcular la complejidad

3.8.2.1. Propuesta 1

Métricas de rendimiento

Medir o estimar el rendimiento de una aplicación en tiempo de diseño es difícil e impreciso [21]. Usamos las llamadas y solicitudes entre microservicios para estimar el rendimiento. Asumimos que, si hay más llamadas y solicitudes entre los microservicios, entonces la comunicación, la latencia y el tiempo de respuesta de la aplicación se incrementa, por lo tanto, el rendimiento de la aplicación se ve directamente afectado. Lo ideal en una aplicación basada en microservicios sería tener microservicios que no tengan comunicación entre ellos y que trabajen de forma independiente. Por lo tanto, definimos tres métricas [21]:

Llamadas de un microservicio (calls_i): Las llamadas corresponden al número de invocaciones de ms_i a otros microservicios [21].

Solicitudes de un microservicio (request_i): Las solicitudes corresponde al número de invocaciones de otros microservicios a ms_i [21].

Promedio de llamadas de MSBA: Corresponde a la suma de las llamadas entre los microservicios dividida entre el número de microservicios (n) de MSBA. Entonces:

$$Avg. Calls = \frac{1}{n} \sum_{i=1}^n Calls_i \quad (16)$$

Ecuación 16. Llamadas de un microservicio

Complejidad cognitiva de MSBA

Según [21] medir la complejidad es fundamental para desarrollar aplicaciones basadas en microservicios. Si la complejidad es alta, el costo de cambio es mayor, también el costo de implementación y el tiempo de desarrollo se incrementan. Las métricas de complejidad utilizadas se detallan a continuación.

Puntos de historia de usuario del microservicio (P_i): Los puntos de historia de usuario son un punto estimado del esfuerzo necesario para desarrollar la historia del usuario. Los puntos

son un indicador de la velocidad del equipo de desarrollo; por lo tanto, definimos P_i como la sumatoria de los puntos de historia de usuario de ms_i de la siguiente manera: [21]

$$P_i = \sum_{j=1}^m PH_j \quad (17)$$

Ecuación 17. Puntos de historia de usuario del microservicio

Donde: P_i es el total de puntos de historia asociados a ms_i .

m es el número de historias de usuario asociadas al microservicio ms_i .

PH_j corresponde a los puntos de historia estimados por el equipo de desarrollo para j-ésima historia de usuario de ms_i .

Puntos de complejidad cognitiva de MSBA (CxT): Se añadieron puntos según la complejidad del microservicio, sus relaciones y dependencias. Se estimó la dificultad de desarrollar y mantener una aplicación basada en microservicios. Se planteó de acuerdo con los siguientes postulados [21]:

- Se estima la dificultad de desarrollar, mantener y entender una aplicación basada en microservicios.
- El punto de partida fue la estimación de los puntos de historias usuarios que realiza el equipo de desarrollo.
- Se agregan puntos de acuerdo con la complejidad del microservicio, sus relaciones y dependencias.
- Se basa en la complejidad de un grafo y su profundidad [21].
- Se tiene en cuenta un caso base, que corresponde a la menor complejidad; este caso sería una MSBA con un solo microservicio, con una sola historia de usuario y un punto de historia estimado para su desarrollo, que sería el caso más simple de desarrollar. Para este caso $Cx_0 = 2$.
- Corresponde al número de veces que MSBA es más compleja en relación con el caso base. Los puntos cognitivos se incrementan de acuerdo con [21]:
- Los puntos estimados totales para cada microservicio que hace parte de la aplicación [21].
- El número de microservicios que hacen parte de la aplicación [21].
- El número de historias de usuario u operaciones asociadas a cada microservicio [21].
- El número de llamadas - calls (out) y peticiones – request (in) del microservicio [21].

- La profundidad del número de llamadas que hace un microservicio a otros microservicios [21]. Corresponde al número de nodos consecutivos usados en la llamada de un microservicio a otro [21].

Entonces formalmente **CxT** se definió de la siguiente manera [21]:

$$\mathbf{Cx} = \left(\sum_{i=1}^n Cg_i \right) + \text{Max}(P_1, \dots, P_n) + (n * \text{WsicT}) + \left(\sum_{i=1}^n Pf_i \right) + \left(\sum_{i=1}^n LC_i \right) \quad (18)$$

Ecuación 18. Puntos de complejidad cognitiva de MSBA

Donde:

i = i-esimo microservicio

Cg_i = $P_i * (\text{Callsi} + \text{Requesti})$

P_i = Número total de historias de Usuario del i-esimo microservicio (Suma de los puntos de historia de Usuario de cada historia de usuario asociada al microservicio).

Max(P₁, ... , P_n) Máximo **P_i** de MSBA.

n = número de microservicios de MSBA.

WsicT (Granularidad): Mayor WSIC de la aplicación, definida en la siguiente ecuación.

$$\mathbf{WsicT} = \text{Max}(\text{Wsic1}, \text{Wsic2}, \dots, \text{WsicTn}) \quad (19)$$

Ecuación 19. Mayor granularidad

Pf_i: Número de nodos usados secuencialmente desde una llamada que realiza un microservicio a otros microservicios, contados desde el i-ésimo microservicio; Una mayor profundidad implica una mayor complejidad de implementación y mantenimiento de la aplicación.

LC: Interdependencia del microservicio

Cx₀: El caso base en el que la aplicación tiene un microservicio, una historia de usuario con un punto de historia estimado. Entonces:

$$\mathbf{Cg}_1 = 0, \mathbf{Max}(P_1) = 1, \mathbf{n}=1, \mathbf{WsicT}=1, \mathbf{Pf}_1=0, \mathbf{SIY}=0, \text{ y } \mathbf{Cx} = 2 \quad (20)$$

Ecuación 20. Valores del caso base

Por lo tanto:

$$\mathbf{Cx}_0 = 2 \quad (21)$$

Ecuación 21. Caso base

$$CxT = \frac{Cx}{Cx_0} \quad (22)$$

Ecuación 22. Complejidad total del sistema

3.8.2.2. Propuesta 2

La granularidad de un servicio se describe como el tamaño adecuado de un servicio, es evaluada como la razón del cuadrado de número de operaciones de un servicio y el cuadrado del número de mensajes consumidos por el servicio y está dado por [30]:

$$granularidad = \frac{\text{número de operaciones}}{\text{número de mensajes}} = \frac{[O(s)]^2}{[M(s)]^2} \quad (23)$$

Ecuación 23. Granularidad de un servicio

Donde:

$O(s)$: Es el conjunto de operaciones proporcionado por un servicio.

$M(s)$: Es el conjunto de mensajes de todas las operaciones de un servicio.

La complejidad se describe como la dificultad de entender la relación entre los servicios, es evaluado como el número de operaciones y está dada por [30]:

$$complejidad = \frac{\text{granularidad del servicio}}{\text{número de servicio}} \quad (24)$$

Ecuación 24. Complejidad del servicio

$$complejidad = \frac{\sum_{i=1}^{O(s)} (SG(i))^2}{S} \quad (25)$$

Ecuación 25. Complejidad total del sistema

Donde:

S : Número de servicios.

$SG(i)$: Granularidad de i -ésimo servicio.

$O(s)$: Número de operaciones del servicio.

3.8.2.3. Propuesta 3

Proporcionan al diseñador una mejor visión interna, ayudan a que el diseño evolucione a un nivel [32].

Complejidad estructural [32].

$$S(i) = [f_{out}^2(i)] \quad (26)$$

Ecuación 26. Complejidad estructural

Donde:

f_{out} = Expansión de datos del módulo

i (número de módulos que inmediatamente dependen al módulo i)

Complejidad de datos sobre la interfaz interna del módulo [32].

(27)

$$D(i) = \frac{v(i)}{[f_{out}(i) + 1]}$$

Ecuación 27. Complejidad de datos

Donde:

$v(i)$ = es el número de variables que entran o salen del módulo [32].

Complejidad total del sistema [32].

$$CS = \sum_i (S(i) + D(i)) \quad (28)$$

Ecuación 28. Complejidad total del sistema

3.9. JUICIO DE EXPERTOS

Básicamente, el juicio de expertos consiste en pedirle a un grupo de personas su opinión sobre un objeto, herramienta o sustancia. Enseñanza, o su opinión sobre un aspecto en particular [33].

La evaluación de expertos es un método de validación, que se utiliza para probar la confiabilidad de un cuestionario en particular, y para esta verificación es necesario seguir ciertos pasos, como recopilar otras opiniones de personas que tienen un conocimiento

significativo, aceptable y confiable del tema, en nuestro caso optamos por utilizar el conocimiento de 3 expertos en el campo de la ingeniería de desarrollo de Microservicios y desarrollo de software, para asegurar la validez y confiabilidad de este estudio.

3.10. COEFICIENTE ALFA DE CRONBACH

El principal tipo de factor de confianza mencionado en la literatura como alfa, también conocida como Alfa Cronbach este es un parámetro estadístico que mide la confiabilidad, consistencia interna y capacidad de respuesta. Consistencia entre factores internos de medida, si la consistencia interna es pobre, entonces el contenido de los elementos también puede ser, no estoy de acuerdo en que el puntaje general no es la mejor unidad de análisis posible para medidas[34].

$$\alpha = \frac{k}{k-1} \left| 1 - \frac{\sum S_i^2}{S_t^2} \right| \quad (29)$$

Ecuación 29. Coeficiente Alfa de Cronbrach

Donde:

k : Número de ítems del instrumento

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

S_t^2 Varianza total del instrumento

α : Coeficiente de confiabilidad del cuestionario

3.11. METODOLOGÍAS DE INVESTIGACIÓN

3.11.1. Concepto de investigación

La investigación es un conjunto de procesos sistemáticos, críticos y empíricos aplicados para estudiar un fenómeno o problema [36].

3.11.2. Enfoque Cuantitativo

Parte de una idea que va acotándose y, una vez delimitada, se derivan objetivos y preguntas de investigación, se revisa la literatura y se construye un marco o una perspectiva teórica [36]. A partir de las preguntas se generan hipótesis y se identifican variables; Se elabora un plan para su prueba (diseño); variables medidas en un contexto dado; Las medidas obtenidas se analizan

utilizando métodos estadísticos, y se extrae una serie de conclusiones respecto de la o las hipótesis [36].

3.11.3. Enfoque Cualitativo

Los estudios cualitativos pueden desarrollar preguntas e hipótesis antes, durante o después de la recolección y el análisis de los datos [36]. Estas actividades suelen ser útiles, primero, para conocer las preguntas de investigación más importantes; Luego modifíquelo y reaccione a él. El trabajo de investigación se desarrolla dinámicamente en ambas direcciones: entre los hechos y su interpretación, y conduce a un proceso algo "circular" en el que la secuencia no siempre es la misma, pues varía con cada estudio [36].

4. MATERIALES Y MÉTODOS

4.1. TIPO DE INVESTIGACIÓN

4.1.1. Investigación cuali-cuantitativa

El tipo de investigación de este trabajo es cuali-cuantitativa, ya que se va a realizar basándose en una revisión bibliográfica de diferentes elementos de investigación relacionados con la arquitectura basada en microservicios, que permita dotar de un marco teórico con aspectos relacionados con el tema dentro de un campo de estudio, utilizando instrumentos de medida para analizar las relaciones entre variables así aplicar el modelo más óptimo en el desarrollo de un caso práctico de microservicios.

4.2. NIVEL DE INVESTIGACIÓN

4.2.1. Investigación descriptiva

Esta investigación descriptiva se va a utilizar para describir los problemas de calidad que existen dentro del desarrollo de aplicaciones utilizando la arquitectura basada en microservicios.

4.3. DISEÑO DE INVESTIGACIÓN

4.3.1. Diseño narrativo

Se utilizará el diseño narrativo porque permitió ir identificando los problemas de complejidad y dependencia que existen en las arquitecturas basadas en microservicios, así ir definiendo las historias de usuario para realizar el modelo de optimización.

4.4. INSTRUMENTOS DE RECOLECCIÓN DE DATOS

4.4.1. Fichas nemotécnicas

Este instrumento se va a utilizar para anexar los aspectos más importantes de las diferentes fuentes bibliográficas, de esta manera ir construyendo el marco teórico del proyecto de investigación.

4.4.2. Encuesta

El siguiente instrumento tiene como objetivo validar la propuesta del modelo de optimización y el prototipo mediante la aplicación de la encuesta que consta de siete preguntas y una ponderación del uno al cinco.

4.5. TÉCNICA DE ANALISIS DE DATOS

4.5.1. Juicios de expertos

Para validar el modelo de optimización y el prototipo se va a aplicar el instrumento de la encuesta a tres expertos con conocimientos y trayectoria en el tema, así verificar la fiabilidad de la investigación realizada.

4.5.2. Coeficiente alfa de Cronbach

Para analizar y tabular los datos del instrumento aplicado en la técnica de juicio de expertos se va a utilizar el coeficiente alfa de Cronbach para validar la hipótesis planteada y medir la fiabilidad de la propuesta.

4.6. TÉCNICAS DE DESARROLLO DE SOFTWARE

Para este trabajo ha sido necesario utilizar prácticas ágiles como el grupo de trabajo Scrum, que permite la creación de roles especiales para el equipo de desarrollo, además de otras prácticas, como es el Product Backlog que se ha utilizado para poder detallar los requisitos necesarios para implementar el trabajo, el objetivo de las historias de usuario es identificar los requisitos predefinidos y finalmente se utilizará el Sprint para poder detallar el desarrollo del proyecto en iteraciones.

4.6.1. Historias de usuario

Se va a utilizar las historias de usuarios que es una explicación general de las funcionalidades que serán parte del prototipo de la aplicación.

4.6.2. Product Backlog

Se va a utilizar el product backlog conformado por las historias de usuario que representa las funcionalidades de la aplicación y lo vamos a priorizar utilizando la técnica de MosCoW que significa que los requisitos de mayor importancia van a tener una prioridad alta, los requisitos de mediana importancia se van a tener una prioridad media y los requisitos de menor importancia van a tener una prioridad baja.

4.6.3. Release Planning

Para realizar el release planning del prototipo se va a utilizar el sprint uno, el sprint dos y el sprint tres. Estos son los sprints que realmente van a hacer desplegados, la entrega del mismo será el 01 de Agosto del 2022.

4.6.4. Sprint

Se va a utilizar el sprint para dividir por partes el desarrollo del prototipo de la aplicación que consta de tres sprints cada uno con un conjunto de historias de usuarios, así mostrar las iteraciones que va teniendo el prototipo.

4.6.5. Técnica de MoSCoW

A continuación, se describe las tareas utilizadas en la técnica MoSCoW según su categoría: M corresponde a las tareas que se deben realizar primero, S corresponde a las tareas que se deberían incluir al finalizar las tareas anteriores, C corresponde a las tareas que se podrían realizar siempre cuando se haya realizado y W corresponde a las tareas que se podrán incorporar en un futuro. Con lo antes mencionado se utilizará la priorización alta media y para la priorización de las historias de usuario.

4.7. MODELO DE OPTIMIZACIÓN DE DEPENDENCIA Y COMPLEJIDAD

4.7.1. Acoplamiento de MSBA

Para calcular la dependencia absoluta primero definimos las historias de usuario de las funcionalidades del sistema, cada una con su punto de historia según la complejidad de la misma. Se agrupa las historias de usuario según la funcionalidad que va a tener cada microservicio. Utilizando las historias de usuario se define la dependencia que existe entre cada microservicio. Se asocia las de dependencias existentes en cada microservicio. En la ecuación

(3) se define el número de microservicios que va a tener la aplicación.

$$\vec{ADS} = [ADS_1, ADS_2, \dots, ADS_n]$$

Una vez obtenido los valores de dependencia que existe en cada microservicio se aplica la ecuación (4), donde se eleva al cuadrado el valor total de dependencias de cada microservicio, luego se saca la raíz cuadrada de la suma de los valores obteniendo así el valor total de la dependencia.

$$AdsT = \left| \vec{ADS} \right| = \sqrt{ADS_1^2 + ADS_2^2 + \dots + ADS_n^2}$$

4.7.2. Complejidad cognitiva de MSBA

Para realizar el cálculo de la complejidad cognitiva primero definimos las historias de usuario de cada microservicio, definimos los puntos de historia de cada historia de usuario según la complejidad de cada una, de esta manera estimar el esfuerzo necesario de desarrollo, los puntos de historia del microservicio se calculan con la ecuación (17).

$$P_i = \sum_{j=1}^m PH_j$$

Añadimos puntos según la complejidad del microservicio, sus relaciones y dependencias, teniendo en cuenta el caso base que corresponde a la menor complejidad, con una sola historia de usuario y un solo microservicio que viene a ser el caso más simple de desarrollar. En la ecuación (18) reemplazamos los valores anteriormente obtenidos.

$$Cx = \left(\sum_{i=1}^n Cg_i \right) + \text{Max}(P_1, \dots, P_n) + (n * WsicT) + \left(\sum_{i=1}^n Pfi \right) + \left(\sum_{i=1}^n LC_i \right)$$

$$CxT = \frac{Cx}{Cx_0}$$

5. ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS

5.1. MODELO DE OPTIMIZACIÓN PROPUESTO

Basándose en la revisión bibliográfica y a los resultados de la comparación de la aplicación de distintas propuestas a continuación, se presenta el modelo de optimización definido para este caso. Se propuso 3 métricas para el cálculo de la dependencia y 3 métricas para el cálculo de complejidad en arquitectura basadas en microservicios, se va a utilizar las métricas que se observa en la Figura 6, estas métricas son las siguientes: Importancia absoluta del microservicio (AIS), dependencia absoluta del microservicio (ADS) e interdependencia de los microservicios SIY), a partir de estas se calcula el acoplamiento para obtener la dependencia entre microservicios. La falta de cohesión (LC) y el grado de la falta de cohesión (Coh) se refiere a que no existe un alto grado de interdependencia. Las métricas para el cálculo de la complejidad son las siguientes: Métrica de rendimiento, puntos de historias de usuario del microservicio y puntos de complejidad cognitiva de MSBA (Arquitectura basada en microservicios), donde la complejidad es la dificultad de entender, desarrollar y mantener la relación que existe entre los microservicios. Para poder aplicar las métricas antes mencionadas primero se definió las historias de usuario como se observa en la Tabla 6, se priorizó las historias de usuario utilizando la técnica de MoSCoW, se estimó el esfuerzo necesario de cada historia de usuario con la técnica planning poker como se puede observar en la Tabla 7, a continuación, se agrupó las historias de usuario según la funcionalidad en cada microservicio y se asignó las dependencias existentes como se observa en la Tabla 8 y Figura 8 finalmente se realizó la base de datos por servicio como se observa en la Figura 9.

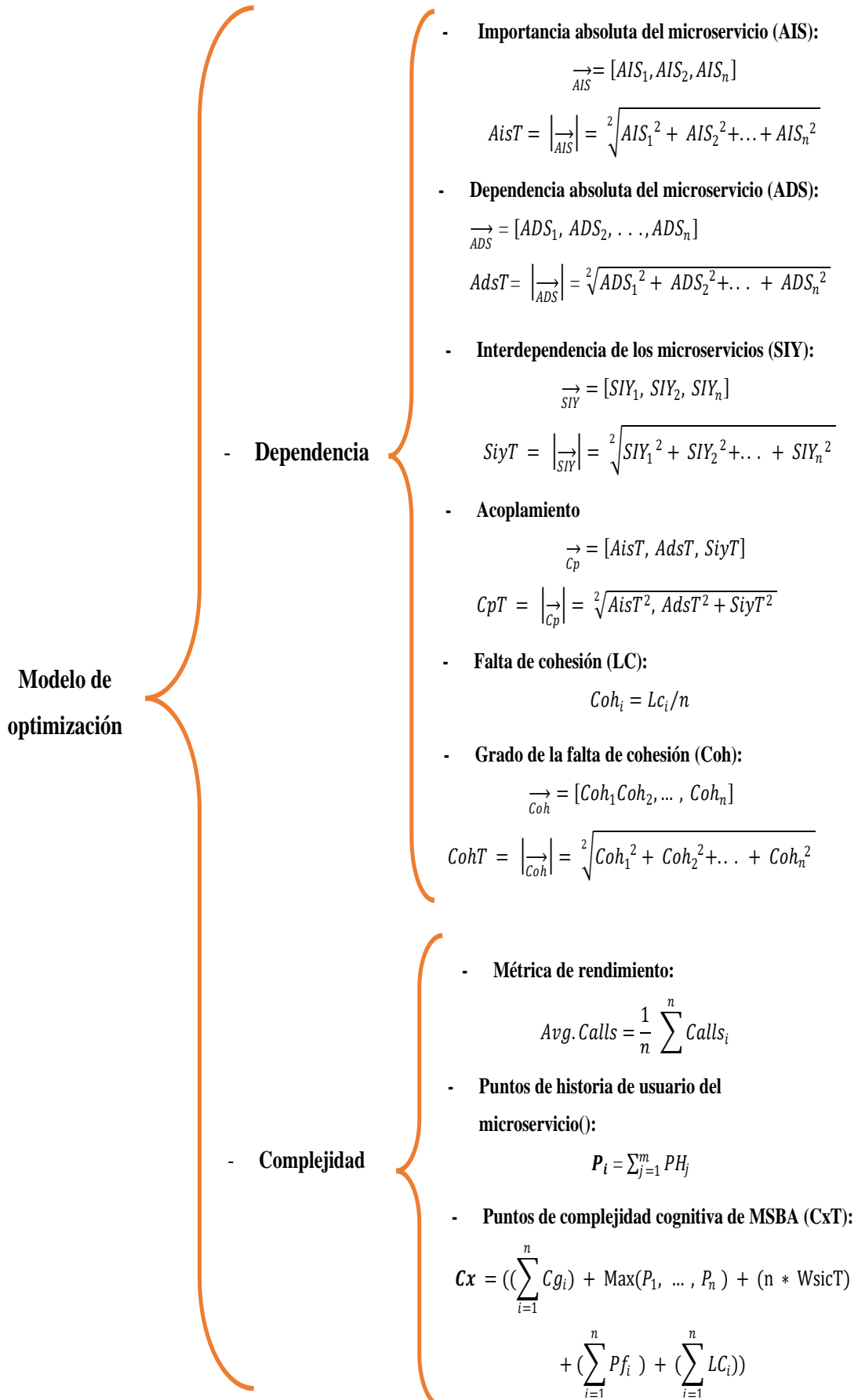


Figura 6. Esquema del modelo de optimización -Dependencia y complejidad

5.2. APLICACIÓN DE LAS DISTINTAS PROPUESTAS DE CÁLCULO DE DEPENDENCIA Y COMPLEJIDAD PARA EL CASO DE ESTUDIO

Caso de estudio: Desarrollo del prototipo de la aplicación de gestión de cursos utilizando el modelo de optimización.

El prototipo de la aplicación web permite que los docentes puedan gestionar cursos, preguntas y exámenes, además asignar a un curso alumnos y exámenes. Los alumnos pueden responder y visualizar las preguntas de los exámenes asignados.

5.2.1. Cálculo de la dependencia

5.2.1.1. Propuesta 1

Datos:

$$AIS_1=1$$

$$AIS_2=0$$

$$AIS_3=1$$

$$AIS_4=1$$

1. En la ecuación (1) se reemplazará los valores de AIS que corresponden a la solicitud de las operaciones según las historias de usuario aplicadas en cada microservicio, como se observa en la Figura 8.

$$\vec{AIS} = [AIS_1, AIS_2, AIS_3, AIS_4]$$

$$\vec{AIS} = [1, 0, 1, 1]$$

2. En la ecuación (2) se eleva al cuadrado los valores de AIS a continuación se suma y se saca la raíz cuadrada para obtener el valor de la importancia absoluta para el caso de estudio.

$$AisT = \left| \vec{AIS} \right| = \sqrt{AIS_1^2 + AIS_2^2 + AIS_3^2 + AIS_4^2}$$

$$AisT = \left| \vec{AIS} \right| = \sqrt{1^2 + 0^2 + 1^2 + 1^2}$$

$$AisT = \left| \vec{AIS} \right| = \sqrt{3}$$

$$AisT = \sqrt{3}$$

$$\mathbf{AisT = 1.73}$$

3. Como resultado se obtuvo 1,73 que corresponde a la importancia absoluta para el caso de estudio.

Dependencia absoluta del microservicio (ADS):

Según [21] menciona que “La dependencia absoluta del microservicio es el número de otros microservicios de los que depende el microservicio”.

Datos:

$$ADS_1=2$$

$$ADS_2=2$$

$$ADS_3=0$$

$$ADS_4=0$$

1. En la ecuación (3) se reemplazará los valores de ADS que corresponden al número de otros microservicios de los que dependen según las historias de usuario aplicadas en cada microservicio, como se observa en la Figura 8.

$$\vec{ADS} = [ADS_1, ADS_2, ADS_3, ADS_4]$$

$$\vec{ADS} = [2, 2, 0, 0]$$

2. En la ecuación (4) se eleva al cuadrado los valores de ADS a continuación se suma y se saca la raíz cuadrada para obtener el valor de la dependencia absoluta del microservicio.

$$AdsT = \left| \vec{ADS} \right| = \sqrt{ADS_1^2 + ADS_2^2 + ADS_3^2 + ADS_4^2}$$

$$AdsT = \left| \vec{ADS} \right| = \sqrt{2^2 + 2^2 + 0^2 + 0^2}$$

$$AdsT = \left| \vec{ADS} \right| = \sqrt{4 + 4 + 0 + 0}$$

$$AdsT = \sqrt{8}$$

$$AdsT = 2.83$$

3. Como resultado se obtuvo 2,83 que corresponde a la dependencia absoluta para el caso de estudio.

Interdependencia de los microservicios (SIY):

Según [21] menciona que “La interdependencia de los microservicios define el número de pares de microservicios que dependen bidireccionalmente uno del otro”.

Datos:

$$SIY_1=0/4$$

$$SIY_2=0/4$$

$$SIY_3=0/4$$

$$SIY_4=0/4$$

1. En la ecuación (5) se reemplazará los valores de SIY que corresponden al número de dependencias bidireccionalmente según las historias de usuario aplicadas en cada microservicio, como se observa en la Figura 8.

$$\vec{SIY} = [SIY_1, SIY_2, SIY_3, SIY_4]$$

$$\vec{SIY} = [0, 0, 0, 0]$$

2. En la ecuación (6) se eleva al cuadrado los valores de SIY a continuación se suma y se saca la raíz cuadrada.

$$SiyT = \left| \vec{SIY} \right| = \sqrt{SIY_1^2 + SIY_2^2 + SIY_3^2 + SIY_4^2}$$

$$SiyT = \left| \vec{SIY} \right| = \sqrt{0^2 + 0^2 + 0^2 + 0^2}$$

$$SiyT = \left| \vec{SIY} \right| = \sqrt{0}$$

$$SiyT = \sqrt{0}$$

$$SiyT = 0$$

3. Como resultado se obtuvo 0 como un buen indicador porque al ser mayor que cero implica mayor complejidad al mantener y desarrollar el caso de estudio.

Métricas de acoplamiento

Según [21] menciona que “El acoplamiento mide el grado de dependencia de un componente de software en relación con otro”.

1. En la ecuación (7) se utilizará los valores de $AdsT$, $AisT$ y $SiyT$ obtenidos anteriormente.

$$\vec{c_p} = [AisT, AdsT, SiyT]$$

$$\vec{c_p} = [1.73, 2.83, 0]$$

2. En la ecuación (8) se eleva al cuadrado los valores de $AdsT$, $AisT$ y $SiyT$ a continuación se suma y se saca la raíz cuadrada

$$CpT = \left| \vec{c_p} \right| = \sqrt{AisT^2 + AdsT^2 + SiyT^2}$$

$$CpT = \left| \vec{c_p} \right| = \sqrt{2.99^2 + 8.01^2 + 0^2}$$

$$CpT = \left| \vec{c_p} \right| = \sqrt{11}$$

$$CpT = 3.32$$

3. Al realizar el cálculo del vector $\left| \vec{c_p} \right|$ se obtiene $CpT = 3.32$ que corresponde al valor del acoplamiento.

Cálculo del grado de la falta de cohesión

Falta de cohesión (LC):

Según [21] menciona que “La falta de cohesión mide el número de pares de microservicios que no tienen ninguna interdependencia (dependencia bidireccional) con el microservicio”.

Datos:

$$LC_1=1/4$$

$$LC_2=0/4$$

$$LC_3=1/4$$

$$LC_4=2/4$$

1. Para el cálculo de falta de cohesión se utilizará la ecuación (9) con los siguientes valores de LC que corresponden al número de dependencias bidireccionalmente según las historias de usuario aplicadas en cada microservicio, como se observa en la Figura 8.

$$Coh_i = Lc_i/n$$

$$Coh_1 = 1/4 = 0.25$$

$$Coh_2 = 0/4 = 0$$

$$Coh_3 = 1/4 = 0.25$$

$$Coh_4 = 2/4 = 0.5$$

2. Como resultado se obtuvo la falta de cohesión de cada microservicio.

Grado de la falta de cohesión (Coh):

Según [21] menciona que “El grado de la falta de cohesión, como la proporción de la métrica de falta de cohesión dividida por el número total de microservicios que forman parte de la aplicación”.

1. En la ecuación (9) se utilizará los valores de Coh obtenidos en el cálculo anterior.

$$\vec{Coh} = [Coh_1, Coh_2, Coh_3, Coh_4]$$

$$\vec{Coh} = [0.25, 0, 0.25, 0.5]$$

2. En la ecuación (10) se eleva al cuadrado los valores de Coh a continuación se suma y se saca la raíz cuadrada.

$$CohT = \left| \vec{Coh} \right| = \sqrt{Coh_1^2 + Coh_2^2 + Coh_3^2 + Coh_4^2}$$

$$CohT = \left| \vec{Coh} \right| = \sqrt{0.25^2 + 0^2 + 0.25^2 + 0.5^2}$$

$$CohT = \left| \vec{Coh} \right| = \sqrt{0.06 + 0 + 0.06 + 0.25}$$

$$CohT = \sqrt{0.37}$$

$$CohT = 0.60$$

3. Como resultado se obtuvo 0.60 que corresponde a la relación que existe en cada microservicio para el caso de estudio.

5.2.1.2. Propuesta 2

Para calcular la segunda propuesta se utilizará la métrica de acoplamiento:

Según [30] menciona que “El acoplamiento de un servicio que describe como la fuerza de la dependencia entre servicios en el sistema”.

Datos:

$$n_c = 4$$

$$n_p = 4$$

$$n_s = 4$$

1. En la ecuación (12) se reemplaza los valores de n_c , n_p y n_s , se divide el número de servicios conectados entre el número de servicios.

$$\text{acoplamiento} = \frac{\text{número de servicios conectados}}{\text{número de servicios}} = \frac{n_c + n_p}{n_s}$$

$$\text{acoplamiento} = \frac{4 + 4}{4}$$

$$\text{acoplamiento} = 2$$

Datos:

$$M(S) = 19$$

$$n_s = 4$$

2. En la ecuación (13) se reemplaza los valores de $M(S)$ y n_s , se divide el número de servicios entre el conjunto de mensajes de las operaciones del servicio para el cálculo de la cohesión.

$$\text{cohesión} = \frac{n_s}{M(S)}$$

$$\text{cohesión} = \frac{4}{19}$$

$$\text{cohesión} = 0.21$$

3. El resultado obtenido de 0.21 corresponde a la cohesión que existe en el caso de estudio.

5.2.1.3. Propuesta 3

Para calcular la tercera propuesta se utilizará la métrica de acoplamiento:

Según [31] menciona que “El acoplamiento es la relación entre dos microservicios puede ser jerárquico (exclusivamente un servicio utiliza los servicios de otro), o bidireccional (ambos servicios utilizan servicios del otro servicio)”.

1. Se obtiene las entradas y salidas de cada historia de usuario, se reemplaza en la ecuación (14) a continuación se suman estos valores por cada microservicio.

$$PF1: (eXit) + (entry) = CFP$$

Datos:

ms_1

$$HU01 = 1(entry) = 1$$

$$HU017 = 1(exit) = 1$$

$$HU019 = 1(exit) = 1$$

$$ms1 = 2(exit) + 1(entry) = 3$$

ms_2

$$HU010 = 1(exit) = 1$$

$$HU011 = 1(exit) = 1$$

$$ms2 = 2(exit) = 2$$

ms_3

$$HU017 = 1(entry) = 1$$

$$ms3 = 1(entry) = 1$$

ms_4

$$HU018 = 2(entry) = 2$$

$$ms4 = 2(entry) = 2$$

2. En la ecuación (15) se reemplaza los valores obtenidos anteriormente y se suman los valores para el cálculo del acoplamiento del sistema.

$$AcpT = ms_1 + ms_2 + \dots + ms_n$$

$$AcpT = ms_1 + ms_2 + ms_3 + ms_4$$

$$AcpT = 3 + 2 + 1 + 2 = 8$$

$$\mathbf{AcpT = 8}$$

3. El resultado obtenido de 8 corresponde al acoplamiento que existe en el caso de estudio.

5.2.2. Cálculo de la Complejidad

5.2.2.1. Propuesta 1

Para calcular la primera propuesta se utilizará las siguientes métricas:

Métricas de rendimiento de MSBA

Según [21] menciona que “Se utiliza las llamadas y solicitudes entre microservicios para estimar el rendimiento”.

1. Para obtener la complejidad del prototipo se deberá calcular las llamadas y solicitudes correspondientes en cada microservicio.

Datos:

Calls	Request
$ms_1=2$	$ms_1=1$
$ms_2=2$	$ms_2=0$
$ms_3=0$	$ms_3=1$
$ms_4=0$	$ms_4=2$

2. En la ecuación (16) se utilizará los valores Calls y Request como se puede observar en la figura 7 donde: $n=4$, $\sum_{i=1}^n Calls_i$ = Suma de las llamadas entre los microservicios.

$$Avg. Calls = \frac{1}{n} \sum_{i=1}^n Calls_i$$

$$Avg. Calls = \frac{1}{n} \sum_{1}^4 4$$

$$Avg. Calls = \frac{1}{4} (16)$$

3. La sumatoria de las llamadas entre los microservicios se divide para n total de microservicios.

$$Avg. Calls = \frac{16}{4}$$

$$Avg. Calls = 4$$

Request = Numero de invocaciones de otro microservicio

$$Request = 4$$

Puntos de historias de usuario del microservicio

1. En la ecuación (17) se utilizará los valores de los puntos de historia utilizando la sucesión de Fibonacci y se realizó la suma total.

$$p_i = \sum_{j=1}^m PH_j$$
$$p_1 = \sum_1^2 1 = 1 + 2 = 3$$
$$p_2 = \sum_1^1 5 = 5$$
$$p_3 = \sum_1^1 8 = 8$$
$$p_i = \sum_3^5 3 + 5 + 8 = 16$$
$$p_i = 16$$

Cálculo de la complejidad cognitiva

1. En la siguiente ecuación se calcula suma de las llamadas y solicitudes entre microservicios donde P_i es la suma de los puntos de historia asociados al prototipo.

$$Cg_i = P_i(calls_i + request_i) = 0$$
$$Cg_i = 16(4 + 4) = 128$$

2. Se agrupa los puntos de historia y se obtiene el número mayor.

$$Max(P_i, \dots P_n)$$
$$Max(8,5,3)$$
$$Max(8)$$

3. En la siguiente ecuación se calcula el mayor valor de historias de usuario asociadas a un microservicio, donde W_{sicT} es la granularidad del microservicio.

Datos:

$$WSIC_1=7$$

$$WSIC_2=7$$

$$WSIC_3=4$$

$$WSIC_4=1$$

$$WsicT = Max(WSIC_1, WSIC_2, WSIC_3, WSIC_4)$$

$$WsicT = Max(7,7,4,1)$$

$$WsicT = Max(7)$$

4. En la siguiente ecuación se calcula el número de microservicios por el número mayor de historias de usuario asociadas a un microservicio.

$$(n * Wsict) = (4 * 7) = 28$$

5. En la siguiente ecuación se calcula el número de llamadas usadas secuencialmente hacia otros microservicios.

$$\Sigma P f_i \rightarrow p f = 0$$

6. Como resultado se obtuvo: historias asociadas: 19, puntos de Historia: 16, total del calls: 4, total del request: 4, número microservicio (n) = 4, wsict = 7, total_calls = 1 y SIY=0, $P_i = 87$.

7. En la ecuación (18) se realiza el cálculo de la complejidad cognitiva con los valores de Cg (llamadas y solicitudes), $Max(P_1, \dots, P_n)$ número mayor de historias de usuario por microservicio, $WsicT$ (Número mayor de Historias de usuario) y Pf (Número de llamadas usados secuencialmente) y LC (Interdependencia del microservicio).

$$Cx = \left(\sum_{i=1}^n Cg_i \right) + Max(P_1, \dots, P_n) + (n * WsicT) + \left(\sum_{i=1}^n P f_i \right) + \left(\sum_{i=1}^n LC_i \right)$$

$$Cx = 128 + 8 + 28 + 0 + 0.60 = 164.60$$

8. En la ecuación (22) se divide la complejidad cognitiva para el caso base $Cx_0 = 2$ citada en el marco teórico.

$$CxT = \frac{Cx}{Cx_0} = \frac{164.60}{2}$$

$$CxT = 82.30$$

9. Implica que para este caso es 82,30 más complejo que el caso base, al tener más historias de usuario implica más operaciones por microservicio y su interfaz es más grande y va a ser más complejo al desarrollar, mantener y probar.

5.2.2.2. Propuesta 2

Según [30] menciona que “La complejidad se describe como la dificultad de entender la relación entre los servicios”

Para calcular la segunda propuesta se utilizará la métrica de complejidad.

Datos:

Calls	Request
$ms_1=2$	$ms_1=1$
$ms_2=2$	$ms_2=0$
$ms_3=0$	$ms_3=1$
$ms_4=0$	$ms_4=2$

1. En la siguiente ecuación se realiza la suma de llamadas y solicitudes para el cálculo de número de mensajes.

$$\text{número de mensajes} = \text{calls} + \text{request}$$

$$\text{número de mensajes} = 4 + 4$$

$$\text{número de mensajes} = 8$$

2. En la ecuación (23) se divide el número de operaciones entre el número de mensajes para obtener el tamaño de microservicio a través de la granularidad.

$$\text{granularidad} = \frac{\text{número de operaciones}}{\text{número de mensajes}} = \frac{[O(s)]^2}{[M(s)]^2}$$

$$granularidad = \frac{[19]^2}{[8]^2}$$

$$granularidad = \frac{361}{64}$$

$$granularidad = SG$$

$$SG = 5,64$$

3. Para el cálculo de la complejidad se utiliza el tamaño del microservicio dividido para el número de servicios que forman parte del sistema.

$$complejidad = \frac{granularidad\ del\ servicio}{número\ de\ servicios}$$

$$complejidad = \frac{\sum_{i=1}^{O(S)} (SG)^2}{S}$$

$$complejidad = \frac{\sum_1^{19} (5,64)^2}{4}$$

$$complejidad = \frac{\sum_1^{19} 31,81}{4}$$

$$complejidad = \frac{603,38}{4}$$

$$complejidad = 151,09$$

4. Como resultado se obtuvo 151.09 que corresponde a la complejidad del caso de estudio.

5.2.2.3. Propuesta 3

Para calcular la tercera propuesta se utilizará las métricas siguientes:

Complejidad estructural

1. En la ecuación (26) se reemplaza los valores $f(i)$ del número de dependencias que tiene el microservicio, a continuación, se eleva al cuadrado para obtener el valor de la complejidad estructural.

$$S(i) = f_{out}^2(i)$$

$$S(i) = 4^2$$

$$S(i) = 16$$

2. Para obtener el número total de variables que salen y entran del microservicio, se suma el total de las llamadas y el total de las solicitudes.

$$v(i) = Tcalls + Trequest$$

$$v(i) = 4 + 4 = 8$$

$$v(i) = 8$$

Complejidad de datos

1. En la ecuación (27) se calcula dividiendo el valor del número total de variables que salen y entran del microservicio entre el número de dependencias que tiene el microservicio.

$$D(i) = \frac{v(i)}{[f_{out}(i) + 1]}$$

$$D(i) = \frac{8}{[16 + 1]}$$

$$D(i) = \frac{8}{17}$$

$$D(i) = 0,47$$

Complejidad total del sistema

1. En la ecuación (28) se suman los valores de la complejidad estructural y la complejidad de datos para el cálculo de la complejidad total del sistema.

$$CS = \sum_i (S(i) + D(i))$$

$$CS = 16 + 0,47$$

$$CS = 16,47$$

2. Como resultado se obtuvo 16,47 que corresponde a la complejidad del caso de estudio.

5.3. COMPARACIÓN DE LOS RESULTADOS DE LAS PROPUESTAS PARA EL CÁLCULO DE DEPENDENCIA Y COMPLEJIDAD

Para cálculo de la dependencia de la primera propuesta se obtuvo los siguientes resultados: Importancia absoluta del caso de estudio con un valor de 1.73, dependencia absoluta de los microservicios con un valor de 2.83, interdependencia de los microservicios 0 estos valores fueron utilizados para obtener el valor de acoplamiento del caso de estudio con un valor de 3.32 correspondientes a bajo acoplamiento. También se calculó el grado de falta de cohesión con un valor de 0.60 correspondientes a bajo grado de falta de cohesión. De la segunda propuesta se obtuvo los siguientes resultados: acoplamiento con un valor de 2 correspondientes a bajo acoplamiento, cohesión con un valor de 0.21 correspondientes a baja cohesión. De la tercera propuesta se obtuvo los siguientes resultados: acoplamiento con un valor de 8 correspondientes a alto acoplamiento.

Para cálculo de la complejidad de la primera propuesta se obtuvo los siguientes resultados:, métricas de rendimiento de MSBA calculando las llamadas con un valor de 4, se utilizó la Tabla 6, Figura 8 con un valor de 4 microservicios, puntos de historia de usuario con un valor de 87, cálculo de la complejidad cognitiva: suma de las llamadas y solicitudes con un valor de 696, máximo valor de puntos de historia con un valor de 8, mayor valor de historias de usuario asociadas a un microservicio con un valor de 7, grado de falta de cohesión con un valor de 0.60 estos valores fueron utilizados para obtener el valor de complejidad del caso de estudio con un valor de 82,30 correspondientes a una baja complejidad. De la segunda propuesta se obtuvo los siguientes resultados: número de mensajes con un valor de 8, granularidad, historias de usuario que puede tener el microservicio con un valor de 5.64, estos valores fueron utilizados para obtener el valor de complejidad con un valor de 151.09 correspondientes a una alta complejidad. De la tercera propuesta se obtuvo los siguientes resultados: complejidad estructural con un valor de 16, suma total de llamadas y solicitudes con un valor de 8, el número de dependencias que tiene el microservicio con un valor de 0.47, estos valores fueron utilizados para obtener el valor de la complejidad total del sistema con un valor de 16.47 correspondiente a una complejidad baja.

Tabla 5. Comparación de las métricas calculadas

Propuesta 1	Propuesta 2	Propuesta 3
DEPENDENCIA		
3,32	2	8
COMPLEJIDAD		
82,30	151,09	16,47

En la Tabla 5 se muestra, la comparación de los resultados del cálculo de dos características de calidad que se detallan a continuación: en la primera propuesta se obtuvo 3.32 y 82.30, en la segunda propuesta se obtuvo 2 y 151.09 y en la tercera propuesta se obtuvo 8 y 16.47 como resultado de dependencia y complejidad. En la segunda propuesta el valor de complejidad es muy alto y su cálculo se realiza de forma general, en la tercera propuesta el valor de la dependencia es alto, esto implica que un microservicio no puede funcionar sin otro microservicio. Por esta razón, el modelo de optimización se realizará basándose en la primera propuesta, obteniendo un bajo acoplamiento, esto quiere decir que no existe mayor dependencia entre microservicios para llevar a cabo una tarea específica, bajo grado de falta de cohesión, esto quiere decir que no existe una dependencia bidireccional entre microservicios y baja complejidad se refiere a que el sistema es fácil de entender desarrollar y realizar cambios sin afectar a las funcionalidades de otros microservicios así su diseño se proyectara con la garantía de optimización de los parámetros de dependencia y complejidad.

5.4. DISEÑO DEL MODELO ARQUITECTÓNICO

Para el caso de estudio, se utilizó las siguientes prácticas ágiles: historias de usuario definidas en la Tabla 6, para realizar la estimación de puntos de historia, se utilizó planning poker que se puede observar en la Figura 6, para priorizar las historias de usuario de cada microservicio se utilizó la técnica de MoSCoW que se puede observar en la Tabla 7, la agrupación y asignación de las historias de usuario por funcionalidad en cada microservicio se puede observar en la

Figura 8. Basándose en estas prácticas se realizó el cálculo de las propuestas investigadas para definir el modelo más óptimo realizando una tabla comparativa de resultados, con el valor de acoplamiento se va a utilizar un tipo de datos para cada microservicio que garantizará que los microservicios estén menos acoplados.

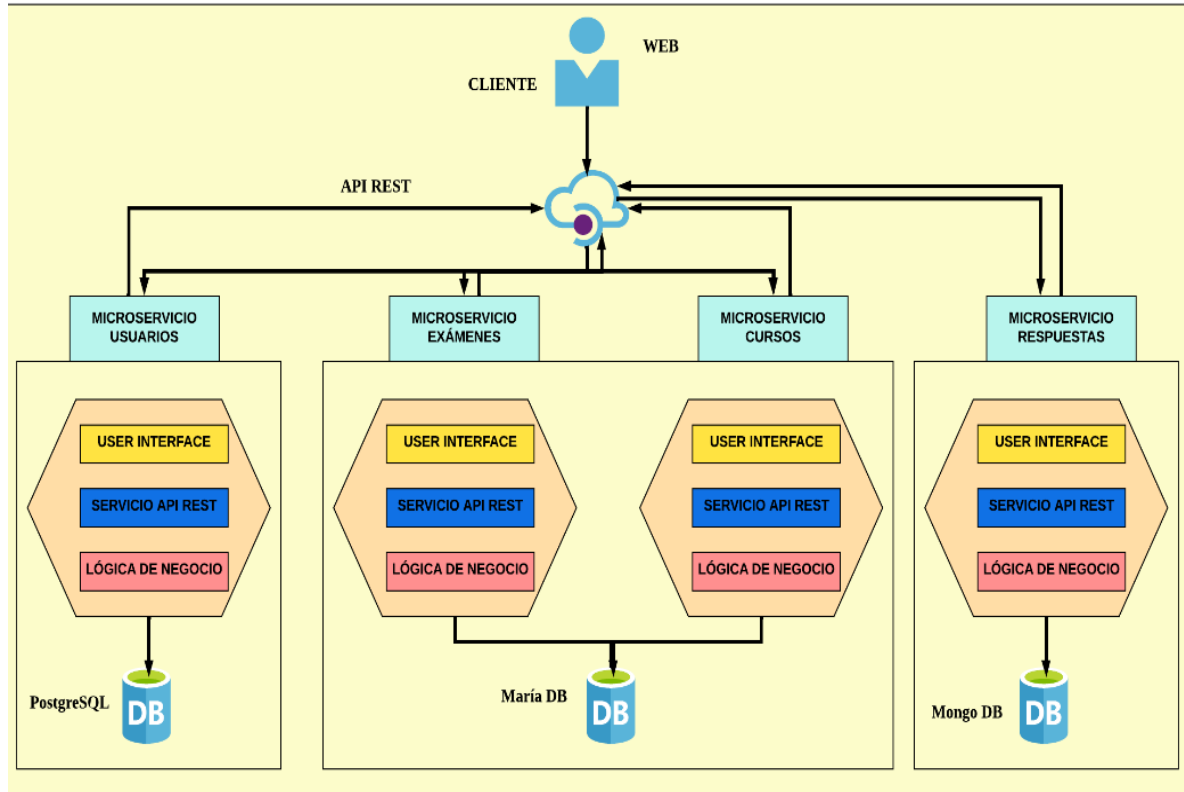


Figura 7. Modelo del diseño de la aplicación

Una vez definido el modelo de optimización se procede al diseño del modelo arquitectónico compuesto de cuatro microservicios: usuarios, exámenes, cursos y respuestas, la API Gateway permite la comunicación entre microservicios, el almacenamiento de información de forma independiente permite que el cambio en la base de datos de un microservicio no afecte a otro, por lo que se va a trabajar con tres bases de datos: relacionales (Mysql, MariaDB) y no relacional (MongoDB) como se observa en la Figura 10.

5.5. IMPLEMENTACIÓN

5.5.1. Historias de usuario

El resultado de los requisitos o funcionalidades del prototipo de la aplicación, como se observa en la Tabla 5.

Tabla 6. Historias de usuario

Identificador de la Historia de Usuario	Descripción de Historias de Usuario
HU001	Como docente requiero registrar un alumno para ser asignado a un curso
HU002	Como docente requiero mostrar los datos del alumno para obtener una lista de alumnos existentes
HU003	Como docente requiero modificar los datos del alumno para tener la información más íntegra posible
HU004	Como docente requiero dar de baja a un alumno para un mejor control de datos
HU005	Como docente requiero dar de baja a un alumno asignado a un curso para un mejor control de datos
HU006	Como docente requiero crear un curso para asignar exámenes y alumnos
HU007	Como docente requiero mostrar un curso para obtener un listado
HU008	Como docente requiero modificar un curso para tener la información más íntegra posible
HU009	Como docente requiero dar de baja un curso para un mejor control de datos
HU010	Como docente requiero asignar un alumno a un curso para que tenga acceso a sus exámenes
HU011	Como docente requiero asignar un examen a un curso para que los alumnos puedan responder
HU012	Como docente requiero crear un examen para asignar a los cursos creados
HU013	Como docente requiero mostrar la lista de exámenes para asignar a un curso específico
HU014	Como docente requiero modificar un examen para tener la información más íntegra posible
HU015	Como docente requiero dar de baja un examen para un mejor control de datos
HU016	Como docente requiero dar de baja un examen asignado a un curso para un mejor control de datos
HU017	Como alumno requiero ver los exámenes respondidos para observar las respuestas
HU018	Como alumno requiero responder el examen para que se guarde y se pueda visualizar las respuestas
HU019	Como alumno deseo ver los exámenes asignados para poder responder

5.5.2.Product Backlog de la aplicación

En la Tabla 6, se muestra el resultado de la priorización de las historias de usuario en el Product Backlog, donde se estableció que historias de usuario tienen una prioridad alta, media y baja, utilizando los puntos de historia para estimar el esfuerzo de las mismas, además se agrupó las historias de usuario en tres sprints.

Tabla 7. Priorización del Product Backlog

Orden de Historia de Usuario Designada	Descripción de Historias de Usuario	Prioridad	Sprint	Puntos de historia
HU006	Como docente requiero crear un curso para asignar exámenes y alumnos	Alta	1	8
HU007	Como docente requiero mostrar un curso para obtener un listado	Alta	1	8
HU008	Como docente requiero modificar un curso para tener la información más íntegra posible	Alta	1	8
HU009	Como docente requiero dar de baja un curso para un mejor control de datos	Alta	1	8
HU012	Como docente requiero crear un examen para asignar a los cursos creados	Media	1	5
HU013	Como docente requiero mostrar la lista de exámenes para asignar a un curso específico	Media	1	5
HU014	Como docente requiero modificar un examen para tener la información más íntegra posible	Media	2	5
HU015	Como docente requiero dar de baja un examen para un mejor control de datos	Media	2	5
HU016	Como docente requiero dar de baja un examen asignado a un curso para un mejor control de datos	Media	2	5
HU001	Como docente requiero registrar un alumno para ser asignado a un curso	Baja	2	3
HU002	Como docente requiero mostrar los datos del alumno para obtener una lista de alumnos existentes	Baja	2	3
HU003	Como docente requiero modificar los datos del alumno para tener la información más íntegra posible	Baja	2	3

HU004	Como docente requiero dar de baja a un alumno para un mejor control de datos	Baja	3	3
HU005	Como docente requiero dar de baja a un alumno asignado a un curso para un mejor control de datos	Baja	3	3
HU010	Como docente requiero asignar un alumno a un curso para que tenga acceso a sus exámenes	Baja	3	3
HU011	Como docente requiero asignar un examen a un curso para que los alumnos puedan responder	Baja	3	3
HU017	Como alumno requiero ver los exámenes respondidos para observar las respuestas	Baja	3	3
HU019	Como alumno deseo ver los exámenes asignados para poder responder	Baja	3	3
HU018	Como alumno requiero responder el examen para que se guarde y se pueda visualizar las respuestas	Baja	3	3
Total Puntos de Historia			87	

5.5.3. Asignación de las historias de usuario

En la Tabla 8, se muestra el resultado de las historias de usuario con las dependencias que existe en cada uno de los microservicios que se utilizó en el modelo de optimización.

Tabla 8. Asignación de dependencia según las historias de usuario por microservicio

Historia de Usuario	Dependencias	Historia de Usuario	Dependencias
HU01	{}	HU11	{HU12}
HU02	{}	HU12	{}
HU03	{}	HU13	{}
HU04	{}	HU14	{}
HU05	{}	HU15	{}
HU06	{}	HU16	{}
HU07	{}	HU17	{HU18}
HU08	{}	HU18	{}
HU09	{}	HU19	{HU18}
HU10	{HU01}		

En la Figura 8, se muestra el resultado gráfico de la agrupación, asignación y dependencias de las historias de usuario que existen entre los cuatro microservicios. El microservicio usuarios compuesto por siete historias de usuario y tres dependencias, el microservicio curso compuesto por siete historias de usuario y dos dependencias, el microservicio exámenes compuesto por cuatro historias de usuario y una dependencia, finalmente el microservicio respuestas compuesto por una historia de usuario y dos dependencias.

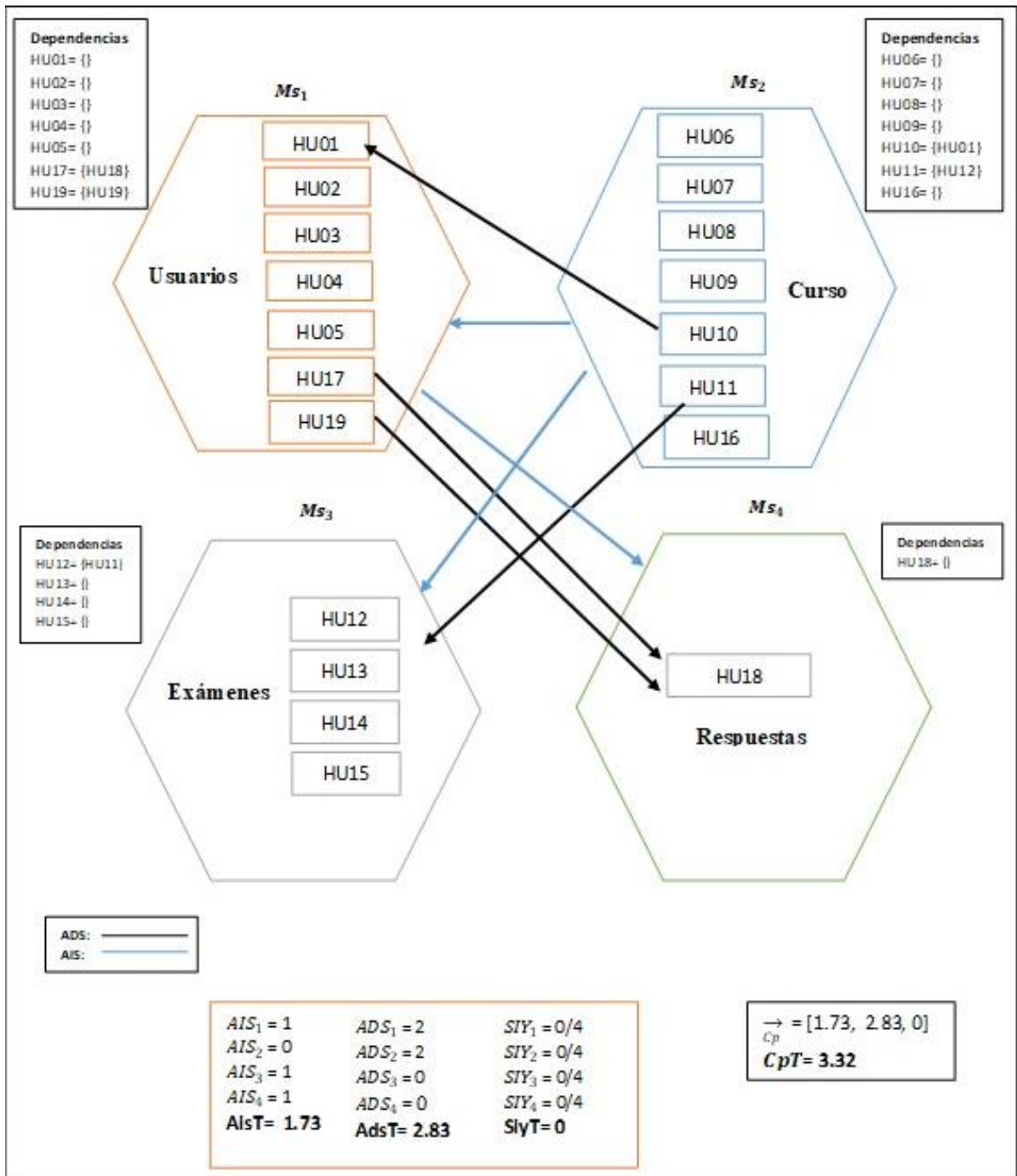


Figura 8. Microservicios, asignación de historias de usuario

En la Figura 9, se observa las llamadas y las solicitudes que se detallan a continuación: el microservicio uno tiene dos llamadas y una solicitud, el microservicio dos tiene dos llamadas y no tiene solicitudes, el microservicio tres no tiene ninguna llamada y tienen una solicitud, finalmente el microservicio cuatro llamadas tiene cero y tiene dos solicitudes.

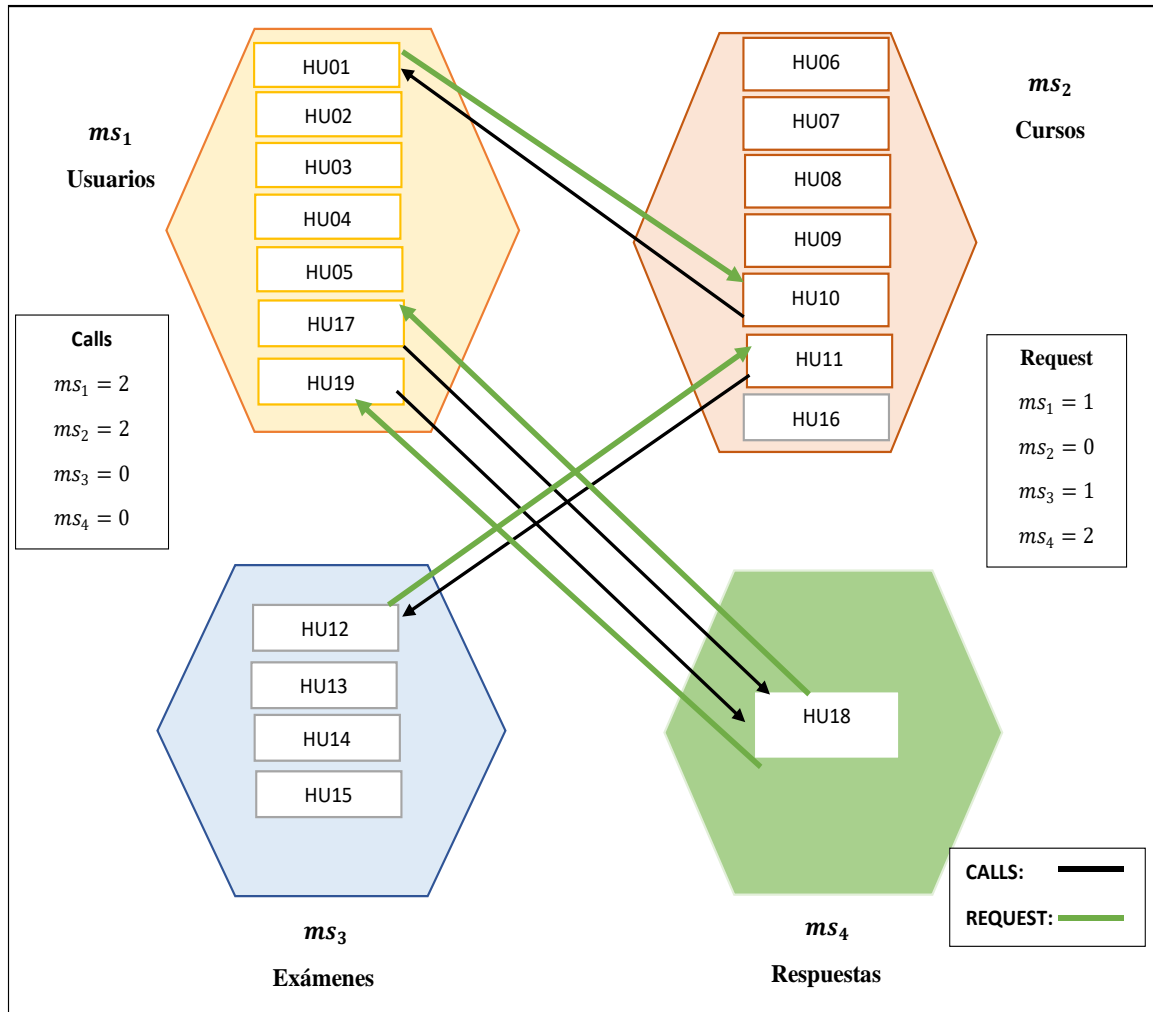


Figura 9. Calls y Request

5.5.4. Base de datos por servicio

En la Figura 9, se muestra el resultado del diseño de las bases de datos conformado de la siguiente manera: microservicio usuario almacena la información de forma independiente en una base de datos relacional (PostgreSQL), microservicio curso y exámenes tienen una base de datos compartida y almacenan su información en una base de datos relacional (MariaBD) y el microservicio respuesta almacena su información de forma independiente en una base de datos no relacional (MongoDB).

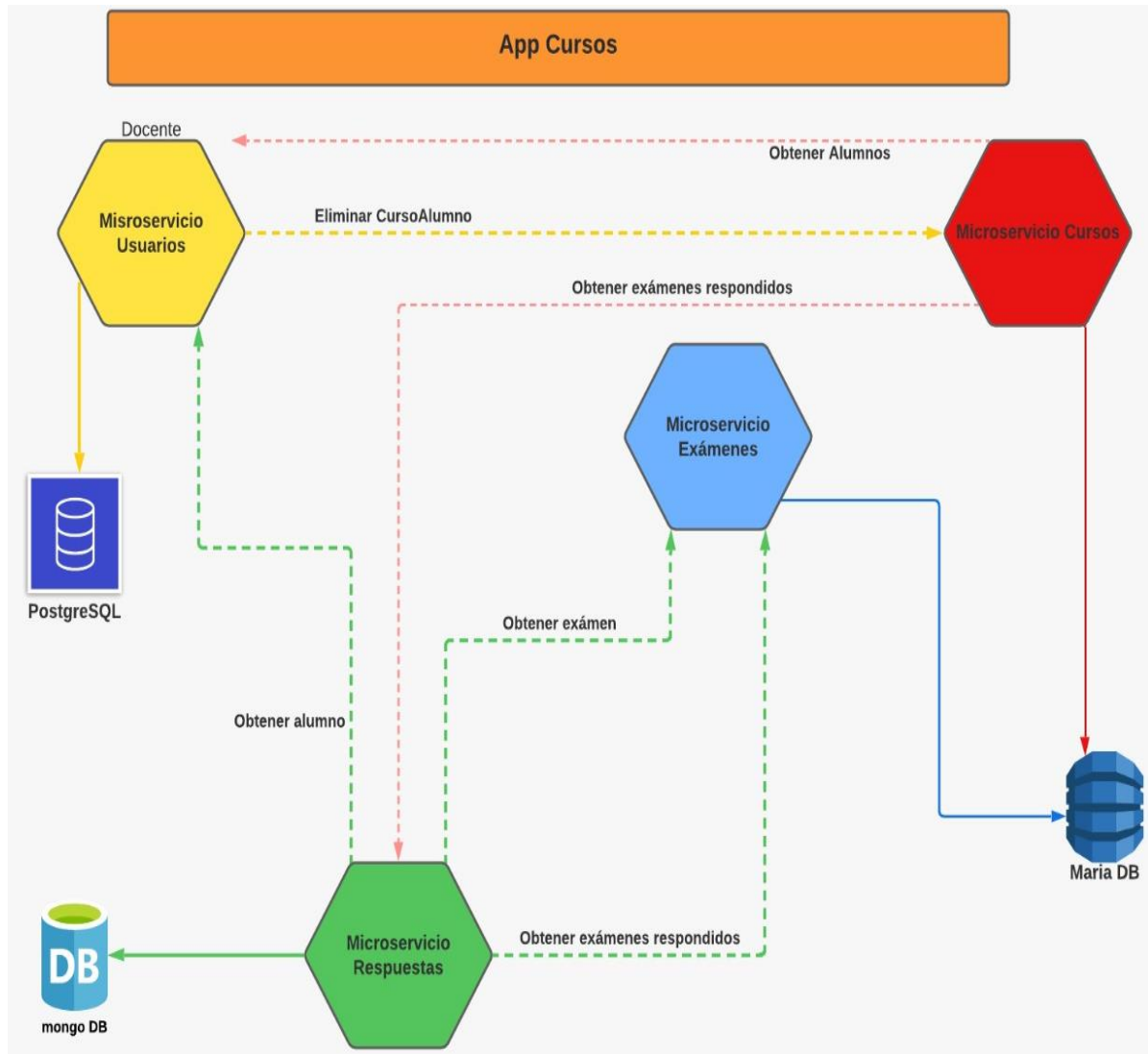


Figura 10. Base de datos por servicio

5.5.5. Ejecución del sprint

5.5.5.1. Detalles de las historias de usuario

A continuación, se presentan los resultados obtenidos del detalle que se realizó con las historias de usuario del primer sprint que corresponden al microservicio Curso, además, se presenta las tareas realizadas como crear, mostrar, actualizar y dar de baja a un curso el cual consta de su descripción, restricción, criterios de aceptación y definition of done, como se puede observar en las Tablas 9 a la 14. (En el ANEXO 2, se describe la arquitectura de todo el proyecto).

Tabla 9. Historia de Usuario HU006 - Crear curso

Historia de Usuario (HU)			
Código HU:	HU006	Fecha:	18/04/2022
Sprint:	1	Prioridad:	Alta
Actor(es):	Docente	Puntos:	8
Descripción:			
Como docente requiero crear un curso para asignar exámenes y alumnos			
Detalle de la HU:			
El Formulario de creación de curso debe contener el siguiente campo: Nombre del Curso			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden acceder a crear un curso.			
Criterios de aceptación:			
1. Dado el formulario de crear un curso, cuando ingresa el nombre del curso. entonces la aplicación creará un curso mostrando el mensaje “Curso ... Creado con éxito” 2. Dado el formulario de crear un curso cuando valide los campos vacíos, entonces la aplicación presentará al usuario el mensaje de error “El campo nombre no debe estar vacío”			
DoD(Definition Of Done):			
La historia debe estar verificada y validada			

Tabla 10. Historia de Usuario HU007 - Mostrar cursos

Historia de Usuario (HU)			
Código HU:	HU07	Fecha:	18/05/2022
Sprint:	1	Prioridad:	Alta
Actor(es):	Docente	Puntos:	8
Descripción:			
Como docente requiero mostrar un curso para obtener un listado			

Detalle de la HU:
El Formulario de mostrar curso debe contener el siguiente campo: Nombre del Curso
Restricciones:
1. Solo los usuarios con privilegios de docente pueden acceder a consultar un curso.
Criterios de aceptación:
1. Dado el formulario de consulta de un curso, cuando el docente haga clic en cursos, entonces la aplicación mostrará la lista de cursos creados
DoD(Definition Of Done):
1. La historia debe estar verificada y validada

Tabla 11. Historia de Usuario HU008 - Modificar curso

Historia de Usuario (HU)			
Código HU:	HU008	Fecha:	18/04/2022
Sprint:	1	Prioridad:	Alta
Actor(es):	Docente	Puntos:	8
Descripción:			
Como docente requiero modificar un curso para tener la información más íntegra posible			
Detalle de la HU:			
El Formulario de modificar un curso debe contener el siguiente campo: Nombre del curso			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden acceder a modificar el nombre del curso.			
Criterios de aceptación:			

<ol style="list-style-type: none"> 1. Dado el formulario que muestre la opción para modificar un curso, cuando se haya modificado el campo del Nombre del curso, entonces la aplicación mostrará el mensaje “Curso ... Actualizado con éxito” 2. Dado el formulario que muestre la opción para modificar un curso, cuando valide los campos vacíos, entonces la aplicación presentará al usuario el mensaje de error “El campo nombre no debe estar vacío”
DoD(Definition Of Done):
La historia debe estar verificada y validada

Tabla 12. Historia de Usuario HU009 - Dar de baja curso

Historia de Usuario (HU)			
Código HU:	HU009	Fecha:	18/05/2022
Sprint:	1	Prioridad:	Alta
Actor(es):	Docente	Puntos:	8
Descripción:			
Como docente requiero dar de baja un curso para un mejor control de datos			
Detalle de la HU:			
El Formulario dar de baja un curso debe contener el siguiente campo: Nombre del curso			
Restricciones:			
<ol style="list-style-type: none"> 1. Solo los usuarios con privilegios de docente pueden acceder a dar de baja un curso. 			
Criterios de aceptación:			
<ol style="list-style-type: none"> 1. Dado el formulario curso que muestre la opción para dar de baja un curso, cuando el docente seleccione un curso, entonces la aplicación emite un mensaje de “Curso Octavo B dado de baja con éxito” 2. Dado el formulario curso que muestre la opción para dar de baja un curso, cuando el docente seleccione un curso, entonces la aplicación mostrará un mensaje de confirmación “¿Seguro que desea dar de baja a Octavo B?” 			
DoD(Definition Of Done):			
<ol style="list-style-type: none"> 1. La historia debe estar verificada y validada 			

Tabla 13. Historia de Usuario HU012 - Crear examen

Historia de Usuario (HU)			
Código HU:	HU012	Fecha:	18/05/2022
Sprint:	1	Prioridad:	Media
Actor(es):	Docente	Puntos:	5
Descripción:			
Como docente requiero crear un examen para asignar a los cursos existentes			
Detalle de la HU:			
El Formulario de creación de exámenes debe contener los siguientes campos: Nombre del examen, Asignatura, Tema, Preguntas			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden crear un examen			
Criterios de aceptación:			
<ol style="list-style-type: none"> 1. Dado el formulario de crear un examen, cuando el usuario registre los datos, Nombre del examen, Asignatura, Tema, Pregunta correctamente, la aplicación emite el mensaje “Examen Creado con éxito”. 2. Dado el formulario de crear un examen, cuando el usuario registre el Tema que exceda en un rango de 3 a 40 caracteres, la aplicación emite el mensaje de error “El tamaño del campo está entre 4 y 30 caracteres”. 3. Dado el formulario de crear un examen, cuando valida los campos y se encuentran vacíos, la aplicación informará al usuario de los posibles errores del formulario 			
DoD(Definition Of Done):			
La historia debe estar verificada y validada			

Tabla 14. Historia de Usuario HU013 - Mostrar exámenes

Historia de Usuario (HU)			
Código HU:	HU013	Fecha:	18/05/2022
Sprint:	1	Prioridad:	Media

Actor(es):	Docente	Puntos:	5
Descripción:			
Como docente requiero mostrar la lista de exámenes para asignar a un curso específico			
Detalle de la HU:			
El Formulario de mostrar la lista de exámenes debe contener los siguientes campos: Nombre del examen, Asignatura, Tema, Preguntas			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden acceder a mostrar la lista de exámenes.			
Criterios de aceptación:			
1. Dado el formulario de exámenes, cuando el docente haga clic en exámenes, entonces la aplicación mostrará la lista de exámenes creados			
DoD(Definition Of Done):			
La historia debe estar verificada y validada			

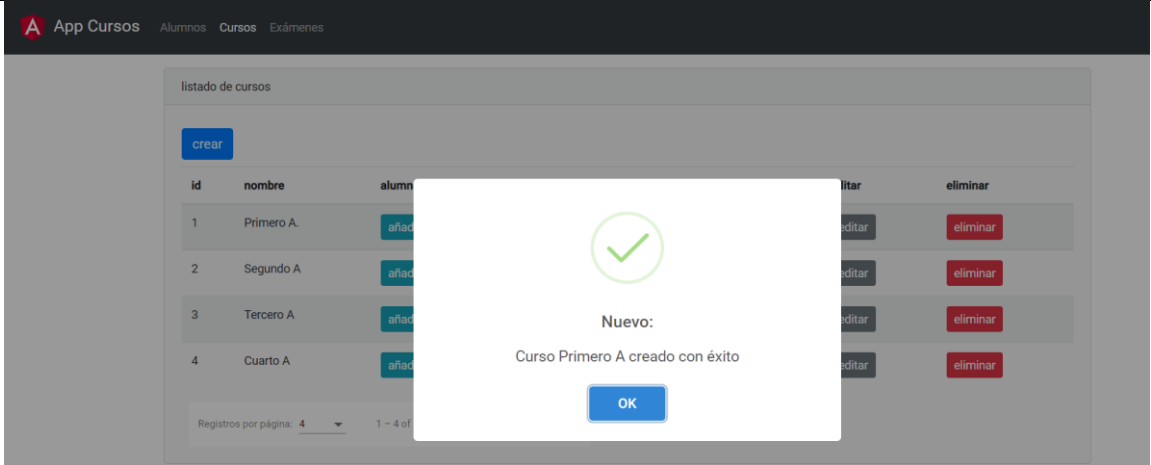
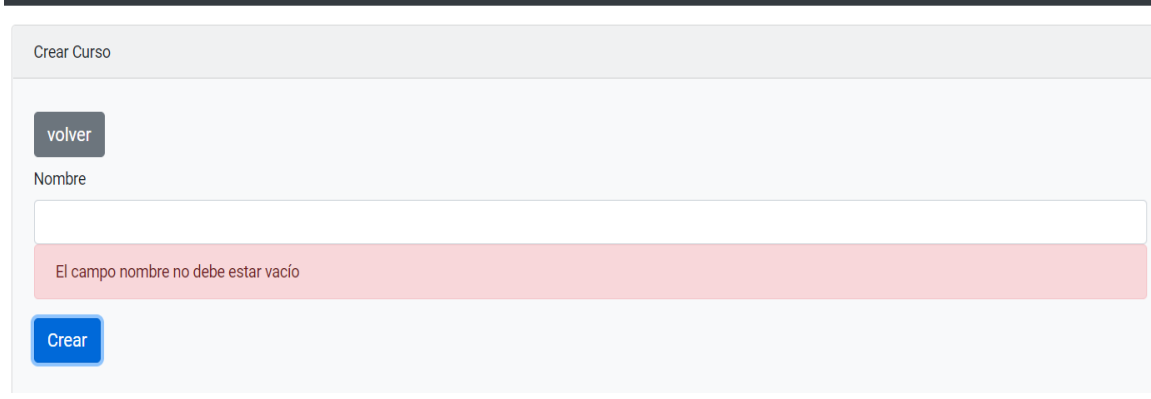
5.5.5.2.Pruebas

Las pruebas que se realizaran a continuación corresponden al primer Sprint “Microservicio cursos”.

Implementación de la historia de usuario “Crear Curso”.

A continuación, se presentan los resultados obtenidos al implementar las funcionalidades y diseño del prototipo basado en arquitectura de microservicio para la creación de un curso, en donde al momento de presionar el botón “CREAR”, el usuario puede observar ingresar el nombre del curso, además la interfaz le muestra un botón que permitirá guardar la información ingresada por teclado en una base de datos, como se puede observar en la Tabla 15.

Tabla 15. Casos de Prueba N 006 - Crear curso

N°:	006	Objetivo:	Crear un curso	Usuario:	Docente
Descripción:	Como docente requiero crear un curso para asignar exámenes y alumnos				
Condiciones de entrada:	<ul style="list-style-type: none"> • Estar autenticado como Docente • Abrir el formulario para agregar el nombre del curso 				
Entrada 1:	Nombre: PRIMERO A				
Resultado esperado:	Curso PRIMERO A creado con éxito	Evaluación de la prueba	Si		
					
Entrada 2:	Si existen campos vacíos				
Resultado esperado:	Mensaje de error “El campo, nombre no debe estar vacío”	Evaluación de la prueba	Si		
					

Implementación de la historia de usuario “Mostrar Cursos”.

A continuación, se presentan los resultados obtenidos al implementar las funcionalidades y diseño del prototipo basado en arquitectura de microservicio para mostrar los cursos, en donde al momento de presionar “Cursos”, el usuario puede ver la información de todos los cursos, como se puede observar en la Tabla 16.

Tabla 16. Casos de Prueba N 007 - Mostrar cursos

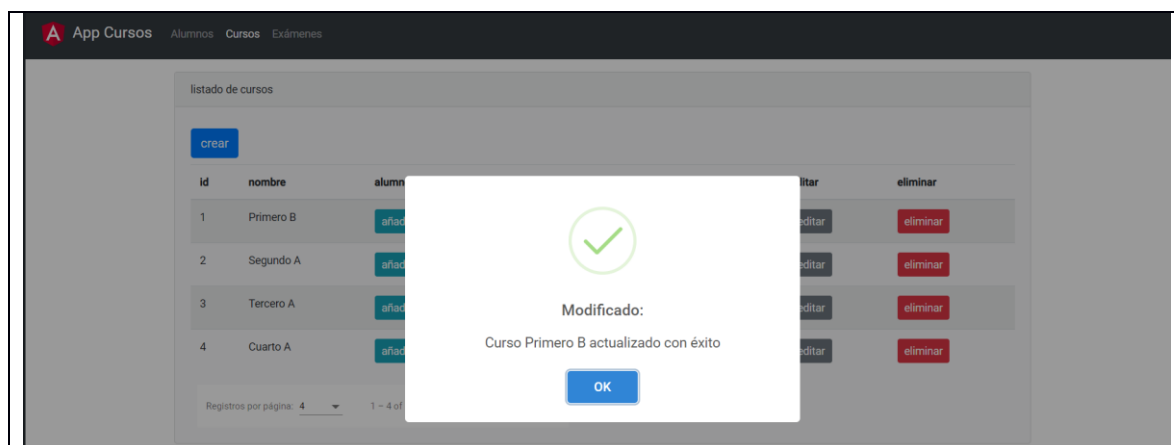
N°:	007	Objetivo:	Mostrar cursos	Usuario:	Docente
Descripción:	Como docente requiero mostrar un curso para obtener un listado				
Condiciones de entrada:	<ul style="list-style-type: none"> • Estar autenticado • Abrir el formulario que muestra todos los cursos 				
Entrada 1:	No aplica				
Resultado esperado:	Muestra la lista de todos los cursos creados	Evaluación de la prueba	Si		

Implementación de la historia de usuario “Modificar Cursos”.

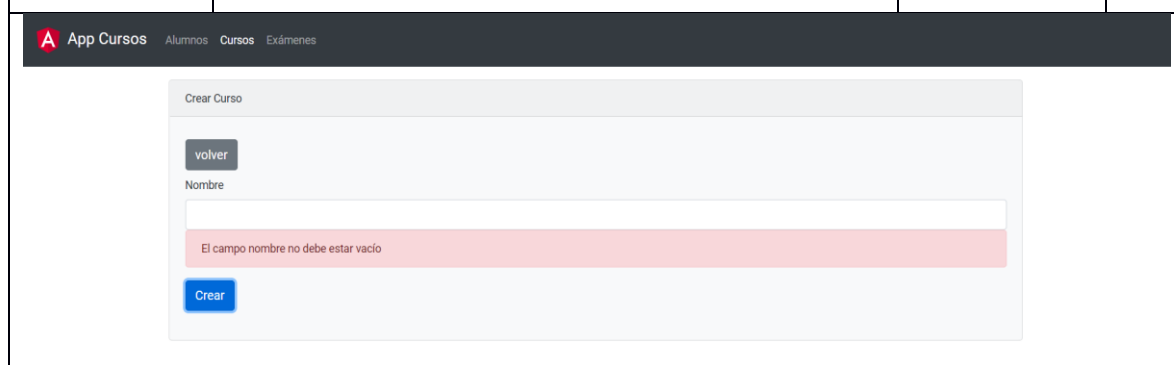
A continuación, se presentan los resultados obtenidos al implementar las funcionalidades y diseño del prototipo basado en arquitectura de microservicio para la creación de un curso, en donde al momento de presionar el botón “Editar”, el usuario puede observar y modificar el nombre del curso, además la interfaz le muestra un botón que permitirá guardar la información modificada por teclado en una base de datos, como se puede observar en la Tabla 17.

Tabla 17. Casos de Prueba N 008 - Modificar cursos

N°:	008	Objetivo:	Modificar un curso	Usuario:	Docente
Descripción:	Como docente requiero modificar el nombre de un curso para tener la información más íntegra posible				
Condiciones de entrada:	<ul style="list-style-type: none"> • Estar autenticado • Abrir el formulario para modificar el nombre del Curso 				
Entrada 1:	Nombre: Primero B,				
Resultado esperado:	Curso Primero B actualizado con éxito	Evaluación de la prueba	Si		



Entrada 2:	Si existen campos vacíos		
Resultado esperado:	Mensaje de error por campos requeridos	Evaluación de la prueba	Si

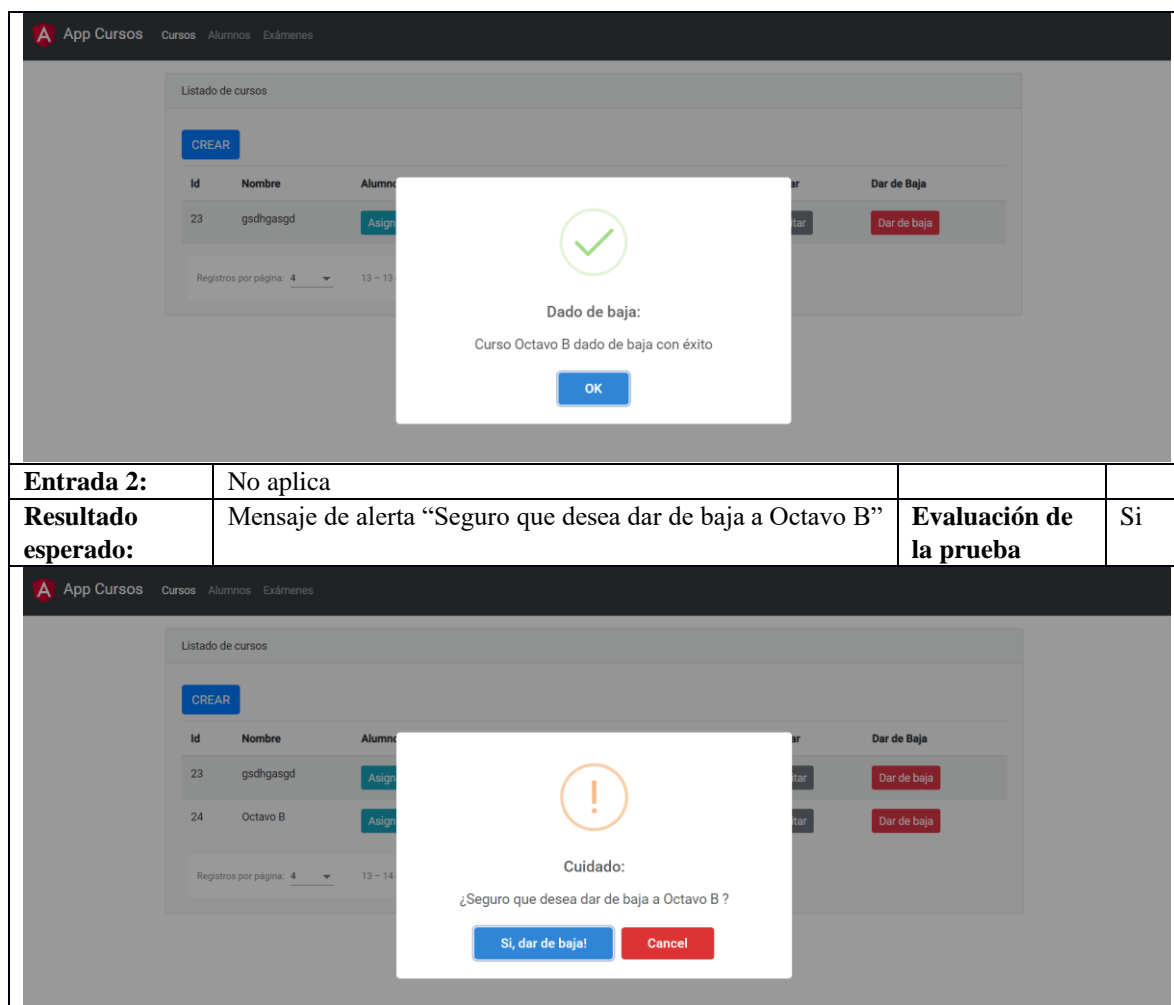


Implementación de la historia de usuario “Dar de Baja Curso”.

A continuación, se presentan los resultados obtenidos al implementar las funcionalidades y diseño del prototipo basado en arquitectura de microservicio para la creación de un curso, en donde al momento de presionar el botón “Dar de baja”, el usuario puede dar de baja un curso, como se puede observar en la Tabla 18.

Tabla 18. Casos de Prueba N 009 - Dar de baja curso

N°:	009	Objetivo:	Dar de baja un curso	Usuario:	Docente
Descripción:	Como docente requiero dar de baja un curso para un mejor control de datos				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Estar autenticado ● Abrir el formulario que muestra la lista de cursos ● Seleccionar el curso 				
Entrada 1:	No aplica				
Resultado esperado:	Mensaje de alerta: “Curso Octavo B dado de baja con éxito”			Evaluación de la prueba	Si



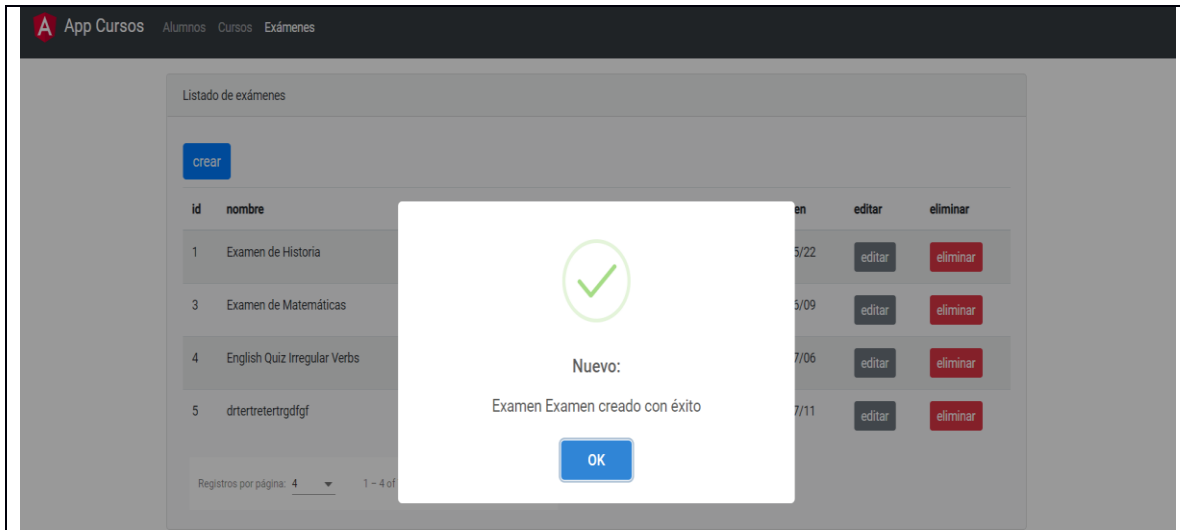
Entrada 2:	No aplica		
Resultado esperado:	Mensaje de alerta “Seguro que desea dar de baja a Octavo B”	Evaluación de la prueba	Si

Implementación de la historia de usuario “Crear Examen”.

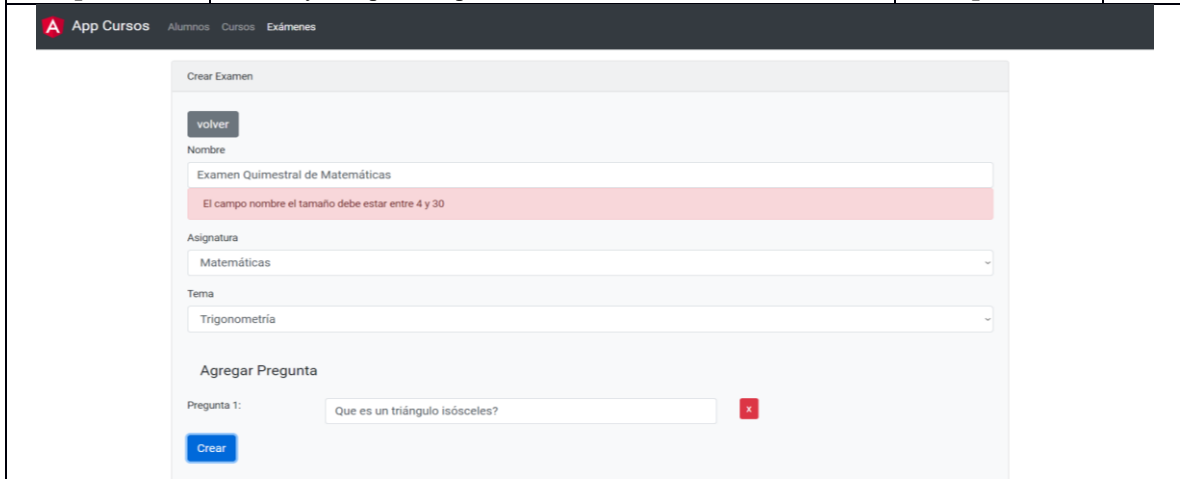
A continuación, se presentan los resultados obtenidos al implementar las funcionalidades y diseño del prototipo basado en arquitectura de microservicio para la creación de un examen, en donde al momento de presionar el botón “Crear”, el usuario puede observar el formulario y llenar los campos, también se presenta el botón Crear que permite guardar la información ingresada por teclado en una base de datos, como se puede observar en la Tabla 19.

Tabla 19. Casos de Prueba N 012 - Crear examen

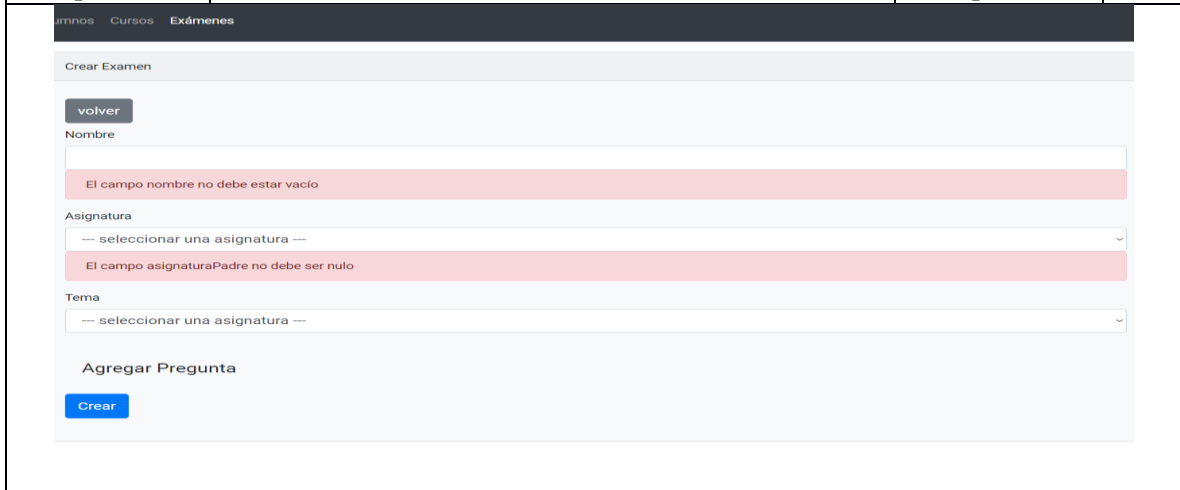
N°:	012	Objetivo:	Crear examen	Usuario:	Docente
Descripción:	Como docente requiero crear un examen para asignar a los cursos creados				
Condiciones de entrada:	<ul style="list-style-type: none"> • Estar autenticado • Abrir el formulario para agregar los datos del examen 				
Entrada 1:	Nombre: Matemáticas, Asignatura: Algebra, Tema: Trigonometría Preguntas: ¿Qué es el triángulo isósceles?				
Resultado esperado:	Examen Matemáticas creado con éxito			Evaluación de la prueba	Si



Entrada 2:	Nombre: Evaluación Quimestral de Matemáticas, Nombre: Matemáticas, Asignatura: Algebra,		
Resultado esperado:	Mensaje de error “ El campo, nombre el tamaño debe estar entre 4 y 30” por campos inválidos	Evaluación de la prueba	Si



Entrada 3:	Si existen campos vacíos		
Resultado esperado:	Mensaje de error por campos requeridos	Evaluación de la prueba	Si



Implementación de la historia de usuario “Mostrar exámenes”.

A continuación, se presentan los resultados obtenidos al implementar las funcionalidades y diseño del prototipo basado en arquitectura de microservicio para mostrar los exámenes, en donde al momento de presionar “Exámenes”, el usuario puede ver la información de todos los exámenes, como se puede observar en la Tabla 20.

Tabla 20. Casos de Prueba N 013 – Mostrar exámenes

N°:	013	Objetivo:	Mostrar exámenes	Usuario:	Docente	
Descripción:	Como docente requiero mostrar la lista de exámenes para asignar a un curso específico					
Condiciones de entrada:	<ul style="list-style-type: none"> • Estar autenticado • Abrir el formulario que muestra todos los cursos 					
Entrada 1:	No aplica					
Resultado esperado:	Muestra la lista de todos los exámenes creados			Evaluación de la prueba	Si	

The screenshot shows the 'App Cursos' interface with a navigation bar containing 'Cursos', 'Alumnos', and 'Exámenes'. The main content area is titled 'Listado de exámenes' and features a 'CREAR' button. Below the button is a table with the following data:

Id	Nombre	Asignaturas	Preguntas	Fecha	Editar	Dar de Baja
1	Examen de Historia	Ciencias Sociales y Historia -> Ciencias Sociales	3	2022/05/22	Editar	Dar de baja
3	Examen de Matemáticas	Matemáticas -> Trigonometría	3	2022/06/09	Editar	Dar de baja
4	English Quiz Irregular Verbs	Inglés -> Gramática	4	2022/07/06	Editar	Dar de baja
10	Gracia	Lenguaje -> Lectura y comprensión	1	2022/07/29	Editar	Dar de baja

At the bottom of the list, there is a pagination control showing 'Registros por página: 4' and '1 - 4 of 4' with navigation arrows.

5.6.VALIDACIÓN

El análisis de Alfa de Cronbach permitió validar el modelo de optimización, la pregunta 1 del instrumento se tomó como referencia para verificar la hipótesis planteada, en la Tabla 30 se puede evidenciar en que rango de confiabilidad se encuentra el valor de 0.72 que se obtuvo del cálculo y análisis de fiabilidad equivalente a excelente confiabilidad, es decir que el modelo de optimización propuesto es eficiente para el diseño de aplicaciones basadas en microservicios.

5.6.1. Resultados de la validación de la propuesta mediante el coeficiente Alfa de Cronbach

A continuación, se muestra los resultados obtenidos del instrumento de validación que está compuesta de 7 preguntas que fueron evaluadas por tres expertos como se puede ver en la Tabla 21.

Tabla 21. Resultados del instrumento de validación aplicado a tres expertos

Experto	Pregunta 1					Pregunta 2					Pregunta 3					Pregunta 4					Pregunta 5					Pregunta 6					Pregunta 7				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Ing. Santiago Merlo			X					X					X					X					X					X					X		
Mg. Diego Muñoz				X				X					X					X					X					X					X		
Ing. Juan Salgado			X				X						X				X						X					X					X		

El experto 1 considero la pregunta 1 con un valor de 3, la pregunta 2 con un valor de 3, la pregunta 3 con un valor de 3, la pregunta 4 con un valor de 4, la pregunta 5 con un valor de 5, la pregunta 6 con un valor de 5 y la pregunta 7 con un valor de 3. El experto 2 considero la pregunta 1 con un valor de 5, la pregunta 2 con un valor de 5, la pregunta 3 con un valor de 4, la pregunta 4 con un valor de 5, la pregunta 5 con un valor de 5, la pregunta 6 con un valor de 5 y la pregunta 7 con un valor de 4. El experto 3 considero la pregunta 1 con un valor de 4, la pregunta 2 con un valor de 2, la pregunta 3 con un valor de 5, la pregunta 4 con un valor de 3, la pregunta 5 con un valor de 4, la pregunta 6 con un valor de 3 y la pregunta 7 con un valor de 4.

A continuación, se detalla cada pregunta de la Tabla 21, basándose en la encuesta aplicada a los expertos.

Tabla 22. Tabulación de la encuesta, pregunta N° 1

	Experto 1	Experto 2	Experto 3
¿ Considera usted que el modelo de optimización propuesto es eficiente para el diseño de aplicaciones basadas en microservicios?	3	5	4

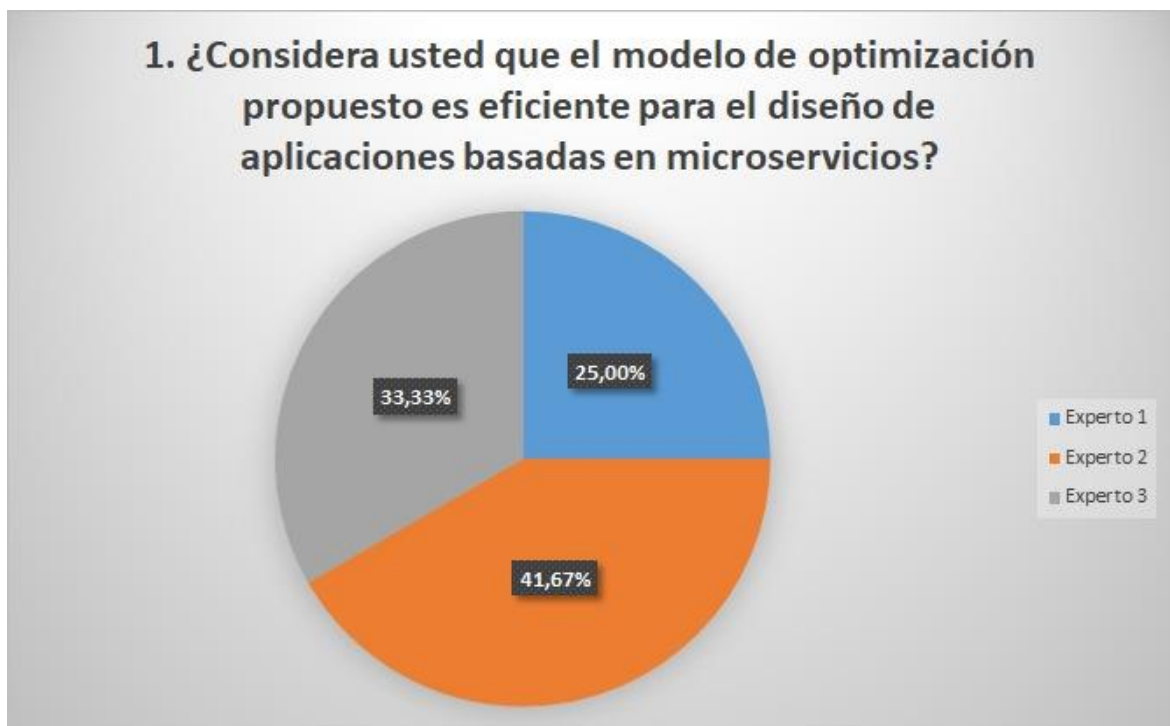


Figura 11. Tabulación pregunta 1

Para la pregunta 1 se obtuvo los siguientes resultados: el primer experto obtuvo 25.0 % equivalente a indiferente, el segundo experto un 41.67 % equivalente a totalmente de acuerdo y el tercero un 33.3 % equivalente a de acuerdo, con esto se puede considerar que el modelo de optimización es eficiente para el diseño de aplicaciones basadas en microservicios.

Tabla 23. Tabulación de la encuesta, pregunta N° 2

Experto 1		Experto 2	Experto 3
¿Considera usted que el modelo de optimización propuesto en este trabajo de investigación ayudaría en el diseño de otras aplicaciones basadas en microservicios?	3	5	2



Figura 12. Tabulación pregunta 2

Para la pregunta 2 se obtuvo los siguientes resultados: el primer experto obtuvo 30.0 % equivalente a indiferente, el segundo experto un 50.0 % equivalente a totalmente de acuerdo y el tercero un 20.0 % equivalente a en desacuerdo, con esto se puede considerar que el modelo de optimización ayudaría en el diseño de otras aplicaciones basadas en microservicios.

Tabla 24. Tabulación de la encuesta, pregunta N° 3

	Experto 1	Experto 2	Experto 3
¿Considera usted que el modelo arquitectónico está de acuerdo al modelo de optimización propuesto?	3	4	5

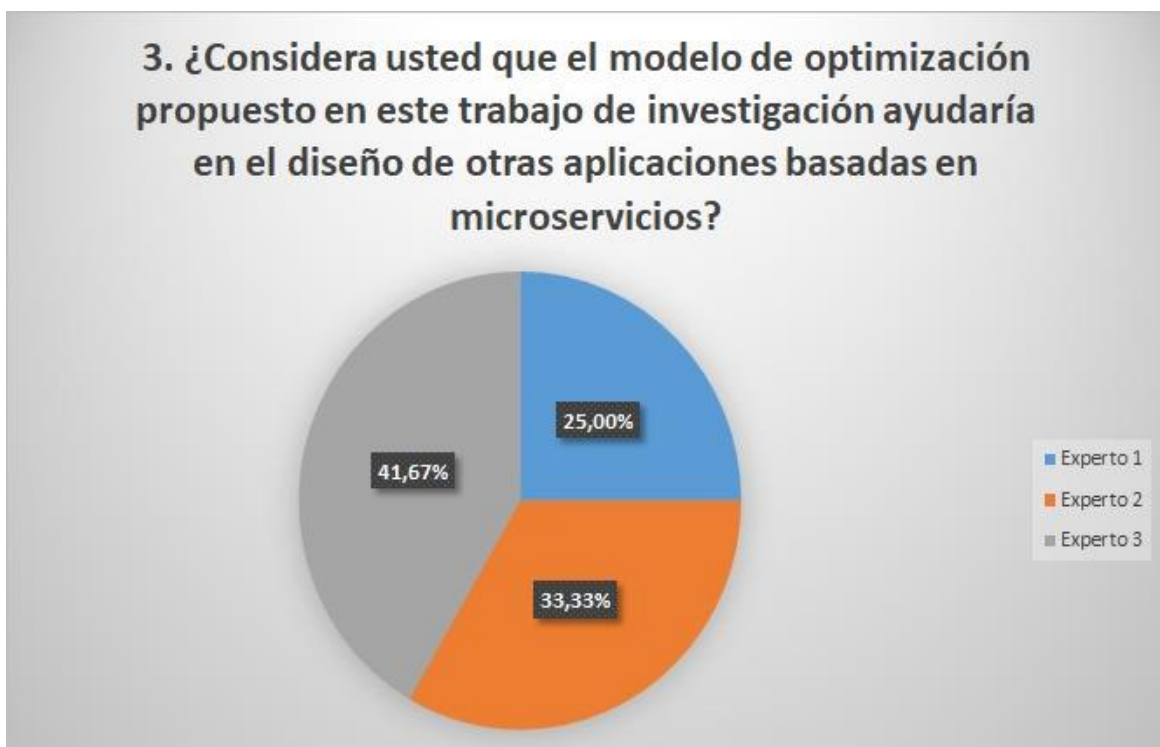


Figura 13. Tabulación pregunta 3

Para la pregunta 3 se obtuvo los siguientes resultados: el primer experto obtuvo 25.0 % equivalente a indiferente, el segundo experto un 33.33 % equivalente a de acuerdo y el tercero un 41.67 % equivalente a totalmente de acuerdo, con esto se puede considerar que el modelo arquitectónico está de acuerdo al modelo de optimización propuesto.

Tabla 25. Tabulación de la encuesta, pregunta N° 4

	Experto 1	Experto 2	Experto 3
¿Considera usted que el prototipo de la aplicación está de acuerdo al modelo arquitectónico ?	4	5	3

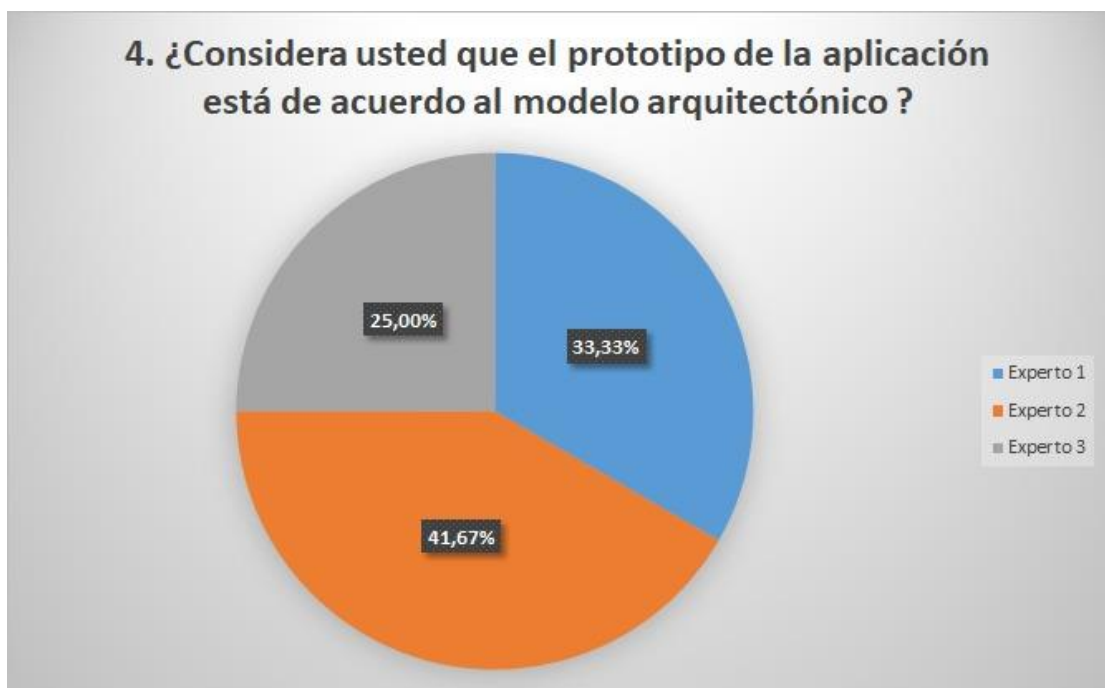


Figura 14. Tabulación pregunta 4

Para la pregunta 4 se obtuvo los siguientes resultados: el primer experto obtuvo 33.33 % equivalente a de acuerdo, el segundo experto un 41,67 % equivalente a totalmente de acuerdo y el tercero un 25.0 % equivalente a indiferente, con esto se puede considerar que el prototipo de la aplicación está de acuerdo al modelo arquitectónico.

Tabla 26. Tabulación de la encuesta, pregunta N° 5

	Experto 1	Experto 2	Experto 3
¿Considera usted que las métricas de importancia absoluta, dependencia absoluta e interdependencia de los microservicios permitieron reducir el acoplamiento?	5	5	4

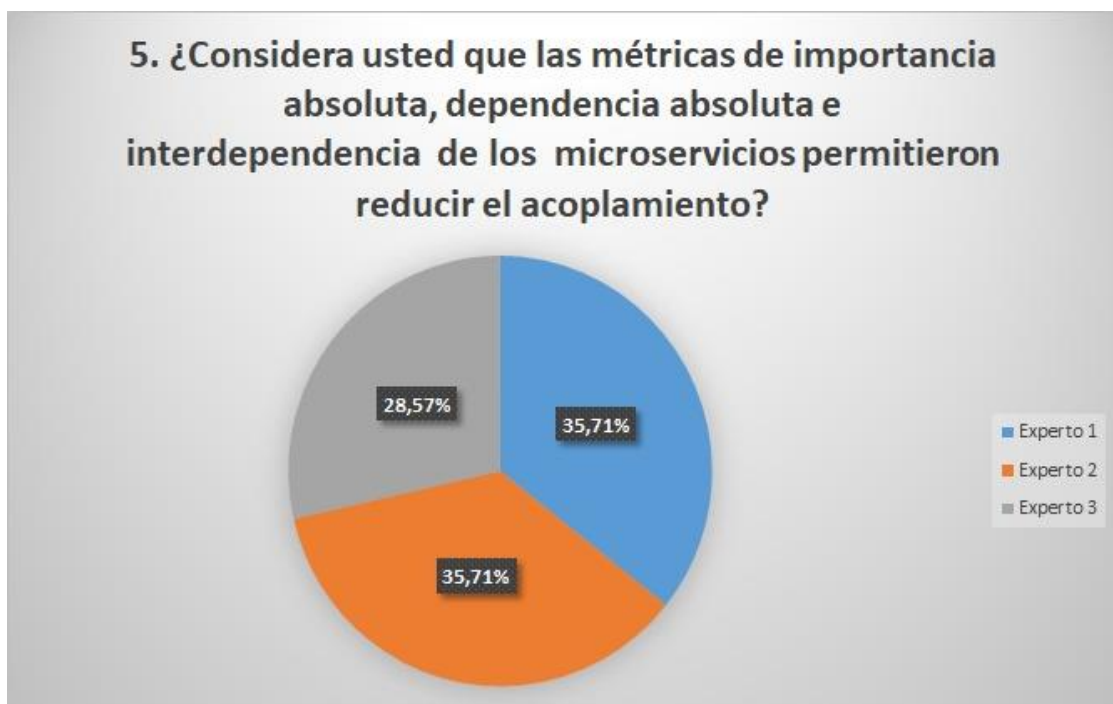


Figura 15. Tabulación pregunta 5

Para la pregunta 5 se obtuvo los siguientes resultados: el primer experto obtuvo 35.71 % equivalente a totalmente de acuerdo, el segundo experto un 35.71 % equivalente a totalmente de acuerdo y el tercero un 28.57 % equivalente a de acuerdo, con esto se puede considerar que las métricas aplicadas en el modelo de optimización permitieron reducir el acoplamiento.

Tabla 27. Tabulación de la encuesta, pregunta N° 6

	Experto 1	Experto 2	Experto 3
¿Considera usted que la métrica de grado de falta de cohesión permitió reducir la dependencia entre los microservicios?	5	5	3



Figura 16. Tabulación pregunta 6

Para la pregunta 6 se obtuvo los siguientes resultados: el primer experto obtuvo 38.46 % equivalente a totalmente de acuerdo, el segundo experto un 38.46 % equivalente a totalmente de acuerdo y el tercero un 23.08 % equivalente a indiferente, con esto se puede considerar que la métrica de grado de falta de cohesión permitió reducir la dependencia entre los microservicios.

Tabla 28. Tabulación de la encuesta, pregunta N° 7

	Experto 1	Experto 2	Experto 3
¿Considera usted que las métricas de rendimiento, los puntos de historia y la complejidad cognitiva permitió reducir la complejidad?	3	4	4

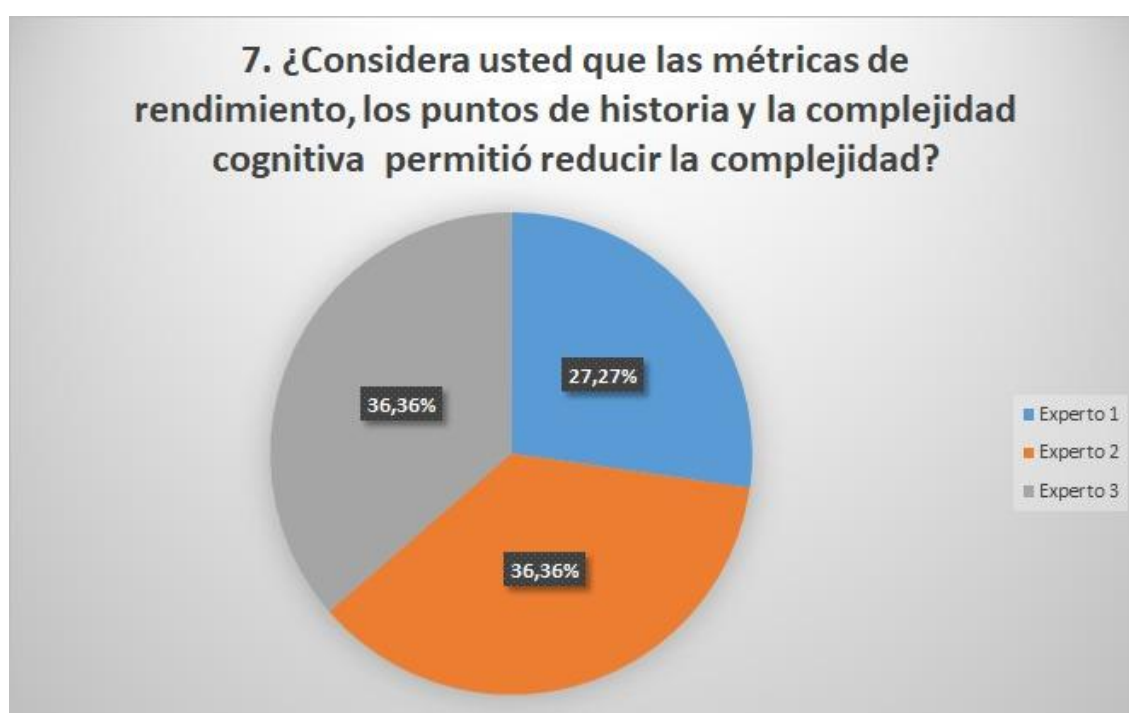


Figura 17. Tabulación pregunta 7

Para la pregunta 7 se obtuvo los siguientes resultados: el primer experto obtuvo 27.27 % equivalente a indiferente, el segundo experto un 36.36 % equivalente a de acuerdo y el tercero un 36.36 % equivalente a de acuerdo, con esto se puede considerar que las métricas aplicadas en el modelo de optimización permitieron la complejidad.

Para obtener los siguientes resultados se utilizó el Alfa de Cronbach aplicando la fórmula establecida, la encuesta está compuesta por 7 preguntas, evaluada por tres expertos en el tema que proporcionaron los datos necesarios para realizar el cálculo y análisis de fiabilidad como se observa en la Tabla 29.

Tabla 29. Tabla estadística del método de Alfa de Cronbach

Expertos Encuestados	1	2	3	4	5	6	7	Sumatoria
E1	3	3	3	4	5	5	3	26
E2	5	5	4	5	5	5	4	33
E3	4	2	5	3	4	3	4	25
Varianza	0.667	1.556	0.667	0.667	0.223	0.889	0.223	
Sumatoria de varianzas	4.889							
Varianza de la suma de los ítems	12.667							

Formula Alfa de Cronbach:

$$\alpha = \frac{k}{k-1} \left| 1 - \frac{\sum S_i^2}{S_t^2} \right|$$

$$\alpha = \frac{7}{7-1} \left| 1 - \frac{4.889}{12.667} \right|$$

$$\alpha = \frac{7}{6} \left| 1 - \frac{4.889}{12.667} \right|$$

$$\alpha = 1.167 | 1 - 0.386 |$$

$$\alpha = 1.167 | 0.614 |$$

$$\alpha = 0.716$$

Donde:

k : Número de ítems del instrumento = 7

$\sum_{i=1}^k S_i^2$: Sumatoria de las varianzas de los ítems = 4.889

S_t^2 Varianza total del instrumento = 12.667

α : Coeficiente de confiabilidad del cuestionario = 0.72

Tabla 30. 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

Como resultado del cálculo del coeficiente de fiabilidad se obtuvo 0.72 que se encuentra en el rango de excelente confiabilidad.

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

5.7.1. Impacto Técnico

Actualmente, el desarrollo de aplicaciones es complejo, la calidad del software se ha convertido en un eje fundamental para su desarrollo, varía según el tamaño y el alto acoplamiento, lo que dificulta su gestión y retrasa la evolución del producto. El modelo de optimización se realizó basándose en las métricas que proporcionan al programador una mejor visión en tiempo de diseño que ayuden a realizar reducir la dependencia y complejidad en cambios a un mayor nivel de calidad de esta mejorar su proceso de desarrollo.

5.7.2. Impacto Social

El modelo de optimización de dependencia y complejidad tiene como objetivo demostrar que aplicando las métricas correctas se puede reducir la dependencia y complejidad en el diseño y desarrollo de aplicaciones basada en arquitectura de microservicios, de esta forma los microservicios funcionarán de forma independiente, pero en conjunto para realizar una tarea

específica mucho más grande, así los futuros desarrolladores podrán aplicar el modelo de optimización antes de desarrollar software ahorrando tiempo y esfuerzo.

5.7.3. Impacto económico

La implementación del modelo de optimización y el prototipo de la aplicación puede causar un impacto económico significativo, ya que al analizar, diseñar, desarrollar e implementar nuevas formas de desarrollo de software se pueden obtener importantes resultados para quienes deciden aprender este tipo de arquitecturas porque su uso es algo nuevo y puede revolucionar el mercado de la tecnología, es posible lograr importantes ingresos económicos, puesto que cada vez más empresas optan por contratar empleados con experiencia en desarrollo de software.

5.8. PRESUPUESTO GENERAL

En esta sección se detallan los gastos directos e indirectos generados en el proyecto de investigación.

5.8.1. Gastos directos

En la Tabla 21, se detalla cada uno de los gastos generados en esta investigación y en el desarrollo del prototipo.

Tabla 31. Estimación de costos – Gastos Directos

Recursos	Cantidad	Valor Unitario, USD \$	Total, USD \$
Internet	2	35.60	71.20
Cursos	2	15.00	30.00
Esferos	2	0.50	1.00
Energía Eléctrica	2	20.00	40.00
Impresiones	350	0.03	10.50
Anillados	4	1.50	6.00
Total, USD \$			158.70

5.8.2. Gastos indirectos

A continuación, se muestra los gastos indirectos generados en este trabajo de investigación y en el desarrollo del prototipo, como se puede observar en la Tabla 22.

Tabla 32. Estimación de costos - Gastos indirectos

Recursos	Días	Valor Unitario, USD \$	Total, USD \$
Transporte	80	0.50	40.00
Alimentación	80	2.50	200.00
Comunicación (Teléfonos)	80	0.50	40.00
Total, USD \$			280.00

5.8.3. Estimación del costo del software por puntos de historia

Para calcular el costo de desarrollo del software correspondiente al prototipo desarrollado en este estudio, se ha utilizado como herramienta principal el método de punto de historia, este método se utilizó para obtener valor en horas de esfuerzo para desarrollar la historia de usuario, luego se presenta el cálculo correspondiente.

Cálculo:

$$Pph = \frac{S}{20hl}$$

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 de días de las historias de usuario

ht = Horas trabajadas

pph = Precio por hora

Reemplazando los datos resulta:

$$pph = \frac{400}{\frac{20}{8}}$$

$$pph = 2.50$$

$$cd = (48 * 6) * 2.50$$

$$cd = \$720$$

Una vez realizados los cálculos, el valor resultante es de \$720, que es el costo total de desarrollo del prototipo.

5.8.4. Gastos generales

En la Tabla 23, se muestra la suma de los gastos directos e indirectos, siendo el costo total de este trabajo de investigación.

Tabla 33. Total de gastos

Gastos directos	\$ 158.70
Gastos indirectos	\$ 280.00
Costo del software	\$ 720
Total	\$ 1158.70

6. CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

- La revisión de fuentes bibliográficas y teorías recopiladas de la literatura, permitieron fundamentar un marco teórico completo que fue la base del proyecto de investigación denominada diseño e implementación de aplicaciones basadas en microservicios utilizando un modelo de optimización de dependencia y complejidad además proporcionaron el conocimiento necesario para la selección de herramientas y metodologías apropiadas para el desarrollo del proyecto de investigación.
- El análisis de las propuestas planteadas de la investigación bibliográfica permitió el modelo más óptimo para reducir la dependencia y complejidad en el diseño de aplicaciones basadas en microservicios.
- El modelo de optimización de dependencia y complejidad propuesto obtuvo un bajo acoplamiento, un bajo grado de falta de cohesión y una baja complejidad cognitiva comparado con las otras propuestas. Usando este modelo de optimización, los desarrolladores de software podrán realizar aplicaciones basadas en microservicios con baja dependencia y baja complejidad, además permitirá agrupar las historias de usuario de distintas maneras para reducir la dependencia, esto en tiempo de diseño.
- Se concluye que el valor obtenido del análisis de fiabilidad del coeficiente Alfa de Cronbach permitió demostrar que el modelo arquitectónico diseñado a través del modelo de optimización propuesto ayudó a reducir de manera considerable la dependencia y complejidad en el diseño para el desarrollo del prototipo de la aplicación.

6.2. RECOMENDACIONES

- Se recomienda investigar más a fondo sobre la dependencia y complejidad en microservicios, debido a que en la actualidad no existe información suficiente en el diseño de aplicaciones basadas en arquitecturas de microservicios.
- Se recomienda analizar otras métricas que ayuden a reducir la dependencia y complejidad en tiempo de diseño para desarrollar aplicaciones menos dependientes y complejas, permitiendo que pueda crecer con mayor facilidad sin afectar a los otros microservicios.

- Se recomienda investigar acerca de las herramientas y tecnologías más utilizadas para desarrollar aplicaciones basadas en microservicios, porque es necesario una constante actualización.
- Se recomienda investigar otras técnicas y métodos que permitan comprobar la hipótesis planteada para medir el grado de fiabilidad del modelo propuesto.

7. BIBLIOGRAFÍA.

- [1] F. H. Vera-Rivera, C. M. G. Cuevas, and H. Astudillo, “Desarrollo de aplicaciones basadas en microservicios: tendencias y desafíos de investigación TT - Development of microservices-based applications: trends and research challenges,” *Rev. Ibérica Sist. e Technol. Informação*, no. E23, pp. 107–120, 2019, [Online]. Available: <http://www.espaciotv.es:2048/referer/secretcode/scholarly-journals/desarrollo-de-aplicaciones-basadas-en/docview/2348878316/se-2?accountid=142712>
- [2] T. I. D. E. Grado, “ARQUITETURA BASADA EN MICROSERVICIOS PARA SISTEMAS DE E-COMMERCE DE LA EMPRESA LCC OPENTECH, C.A,” 2019.
- [3] W. E. B. D. E. La and A. Nacional, “DISEÑO DE UN PROTOTIPO DE UNA ARQUITECTURA BASADA EN MICROSERVICIOS PARA LA INTEGRACIÓN DE APLICACIONES WEB ALTAMENTE TRANSACCIONALES. CASO: ENTIDADES FINANCIERAS,” 2017.
- [4] I. Sommerville, *Ingeniería de software 9*, Perason. Mexico, 2011. [Online]. Available: [Ingenieria de software](#)
- [5] J. L. P. Laje, “ESTUDIO COMPARATIVO ENTRE UNA ARQUITECTURA CON MICROSERVICIOS Y CONTENEDORES DOCKERS Y UNA ARQUITECTURA TRADICIONAL (MONOLÍTICA) CON COMPROBACIÓN APLICATIVA,” 2018, [Online]. Available: [http://repositorio.ug.edu.ec/bitstream/redug/32755/1/B-CISC-PTG-1579 Pacheco Laje José Luis.pdf](http://repositorio.ug.edu.ec/bitstream/redug/32755/1/B-CISC-PTG-1579%20Pacheco%20Laje%20José%20Luis.pdf)
- [6] R. S. Pressman and D. Ph, *Ingeniería del software un enfoque práctico séptima edición*. Mexico.
- [7] B. C. Falgueras, *Ingeniería de Software*. Barcelona, 2003.

- [8] E. G. Maida and J. Pacienza, “Metodologías de desarrollo de software,” 2015.
- [9] O. Blancarte, “Introducción a la arquitectura de software – Un enfoque práctico,” no. 20 20.
- [10] J. del Sagrado, I. M. del Águila, and A. Bosch, “Expansión cuantitativa del método MoSCoW para la priorización de requisitos,” *Actas las 23rd Jornadas Ing. del Softw. y Bases Datos, JISBD 2018*, no. October, 2018.
- [11] N. Cadavid, F. Martínez, J. Daniel, and M. Vélez, “Revisión de metodologías ágiles para el desarrollo de software,” 2013.
- [12] Z. Zhang, “The Benefits and Challenges of Planning Poker in Software Development : Comparison Between Theory and Practice,” 2017.
- [13] R. D. N. y R. D. C. Lozada, “APLICACIÓN BASADA EN ARQUITECTURA DE MICROSERVICIOS,” 2020.
- [14] A. R. Quintero, “Las 4 claves para alcanzar el éxito en la metodología DevOps”.
- [15] A. Guillermo and J. Marco, “DevOps , la nueva tendencia en el desarrollo de sistemas TI , un caso práctico en el análisis de incidencias de software Control Documental,” 2016.
- [16] M. De, M. Callejas-cuervo, A. C. Alarcón-aldana, and A. M. Álvarez-carreño, “Modelos de calidad del software, un estado del arte*,” vol. 13, no. 1, pp. 236–250, 2017.
- [17] I. E. de Normalización, “INGENIERÍA DEL SOFTWARE. CALIDAD DEL PRODUCTO SOFTWARE. mODELO DE CALIDAD (ISO/IEC 9126-1:2001, IDT),” 2014.
- [18] M. Sicilia, “Estándar ISO 9126 del IEEE y la Mantenibilidad,” pp. 2–3, 2009.
- [19] D. Rud, S. Post, and A. Schmietendorf, “Product Metrics for Service-Oriented Infrastructures,” no. May, 2014.
- [20] P. Clements and R. Kazman, “Software Architecture in Practice Addison-Wesley,” 2012.
- [21] F. H. V. R. MSc., “MODELO INTELIGENTE DE ESPECIFICACIÓN DE LA

- GRANULARIDAD DE APLICACIONES BASADAS EN MICROSERVICIOS,” pp. 1–153, 2020.
- [22] S. Hassan, “Microservices and Their Design Trade-Offs : A Self-Adaptive Roadmap Microservices and Their Design Trade-offs : A Self-Adaptive Roadmap,” no. June 2016, 2018, doi: 10.1109/SCC.2016.113.
- [23] M. Lewis, J & Fowler, “Microservices. A definition of this new architectural term. Obtenido de Martinowler.com:,” 25 de Marzo de 2014, 2014. <https://martinfowler.com/articles/microservices.html%0D>
- [24] G. Aguirre, “Sistema de dosificación para una máquina empacadora de snacks,” *Univ. Técnica del Norte*, p. 149, 2018.
- [25] K. Calvo, J. Durán, E. Quirós, E. Malinowski, U. D. C. Rica, and C. Rica, “MongoDB : alternativas de implementar y consultar documentos MongoDB : alternativas de implementar y consultar documentos Introducción,” no. October, 2017.
- [26] P. Martinez, *Desarrollo de aplicaciones mediante el Framework de Spring*. 2015. [Online]. Available: <https://elibro.net/es/lc/utcotopaxi/titulos/107207>
- [27] V. Snapshot, “Tutorial Introducción a Maven 3 Contenido”.
- [28] J. E. Pérez, “Introducción a JavaScript,” 2008.
- [29] I. N. G. Diego and J. Trejo, “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 AUTOR : ANDRÉS RUBÉN GUDIÑO QUINTEROS DIRECTOR :,” 2018.
- [30] D. A. Ruelas, “Modelo de composición de microservicios para la implementación de una aplicación web de comercio electrónico utilizando kubernetes,” *Univ. Nac. del Altiplano*, p. 122, 2017, [Online]. Available: <http://repositorio.unap.edu.pe/handle/UNAP/6781>
- [31] R. Pedraza-coello and F. Valdés-souto, “Measuring Coupling in Microservices Using COSMIC Measurement Method,” no. c, pp. 16–22, 2021.

- [32] R. Pressman, “Métricas Técnicas del Software,” *Ing. Softw.*, pp. 1–37, 2010.
- [33] S. Università, “La validación por juicio de expertos : dos investigaciones cualitativas en Lingüística aplicada Validation by expert judgements : two cases of qualitative research in Applied Linguistics,” 2015.
- [34] J. De Dios, M. Núñez, and A. A. Iniesta, “Culcyt // Instrumentos de medición Coeficiente alfa de Cronbach para medir la fiabilidad de un cuestionario difuso Resumen Introducción,” no. 59, pp. 146–156, 2016.
- [35] S. Panichella, M. Rahman, and D. Taibi, “Structural Coupling for Microservices,” no. March, pp. 280–287, 2021, doi: 10.5220/0010481902800287.

8. ANEXOS

ANEXO 1: INFORME DE PLAGIO

Document Information

Analyzed document	Tesis_Salazar_Toasa.pdf (D143359244)
Submitted	2022-08-29 15:32:00
Submitted by	Verónica Tapia
Submitter email	veronica.tapia@utc.edu.ec
Similarity	3%
Analysis address	veronica.tapia.utc@analysis.arkund.com

Sources included in the report

SA	UTN_MIS_Tesis-DLH.docx Document UTN_MIS_Tesis-DLH.docx (D31736650)	10
SA	HerramientaEvalProd - AngelCalderon.docx Document HerramientaEvalProd - AngelCalderon.docx (D112746060)	2
SA	TTKevinQuito.docx Document TTKevinQuito.docx (D136926534)	1
SA	TorresErazoAngelMauricio.pdf Document TorresErazoAngelMauricio.pdf (D78415347)	1
SA	SUB. TESIS BADILLO_CORREGIDO[19943].docx Document SUB. TESIS BADILLO_CORREGIDO[19943].docx (D117243883)	1
SA	UNIAJC TECNOHELP(AnaCorrecciones) (1).docx Document UNIAJC TECNOHELP(AnaCorrecciones) (1).docx (D107430415)	1
SA	20210713_TG_SGST_V9.0.docx Document 20210713_TG_SGST_V9.0.docx (D110681265)	1
W	URL: https://www.ibm.com/downloads/cas/5O8YOPKN Fetched: 2019-10-22 18:22:12	1

Entire Document

ANEXO 2: ARQUITECTURA DEL PROYECTO

1. TÍTULO DEL SISTEMA

Prototipo de una aplicación web para la gestión de cursos, alumnos y exámenes

2. QUÉ HACE EL SISTEMA

El prototipo de la aplicación web permite que los docentes puedan gestionar cursos, preguntas y exámenes, además asignar a un curso alumnos y exámenes. Los alumnos pueden responder y visualizar las preguntas de los exámenes asignados.

3. OBJETIVOS

Utilizar el modelo matemático para la creación del prototipo de una aplicación web de arquitectura basada en microservicios que permita la optimización de dependencia y complejidad en el diseño del microservicio.

- Definir las historias de usuarios de los diferentes microservicios, para la gestión de alumnos, cursos, exámenes.
- Diseñar el modelo matemático que permita la optimización de dependencia y complejidad en el diseño del microservicio.
- Crear el prototipo utilizando el modelo matemático de optimización de dependencia y complejidad.
- Desarrollar las pruebas del prototipo que permitirán verificar la consistencia en las funcionalidades en el diseño del prototipo de la aplicación web.

4. EMPRESAS A LAS QUE SIRVE

Sector educativo a nivel medio y superior

5. IDEA DEL PROYECTO

En el Gráfico 1, se muestra una idea conceptual general del prototipo, el mismo que contiene las funcionalidades: el docente gestiona cursos, preguntas, exámenes, además asigna a un curso alumnos y exámenes. El alumno puede visualizar las preguntas de los exámenes asignados y responder.

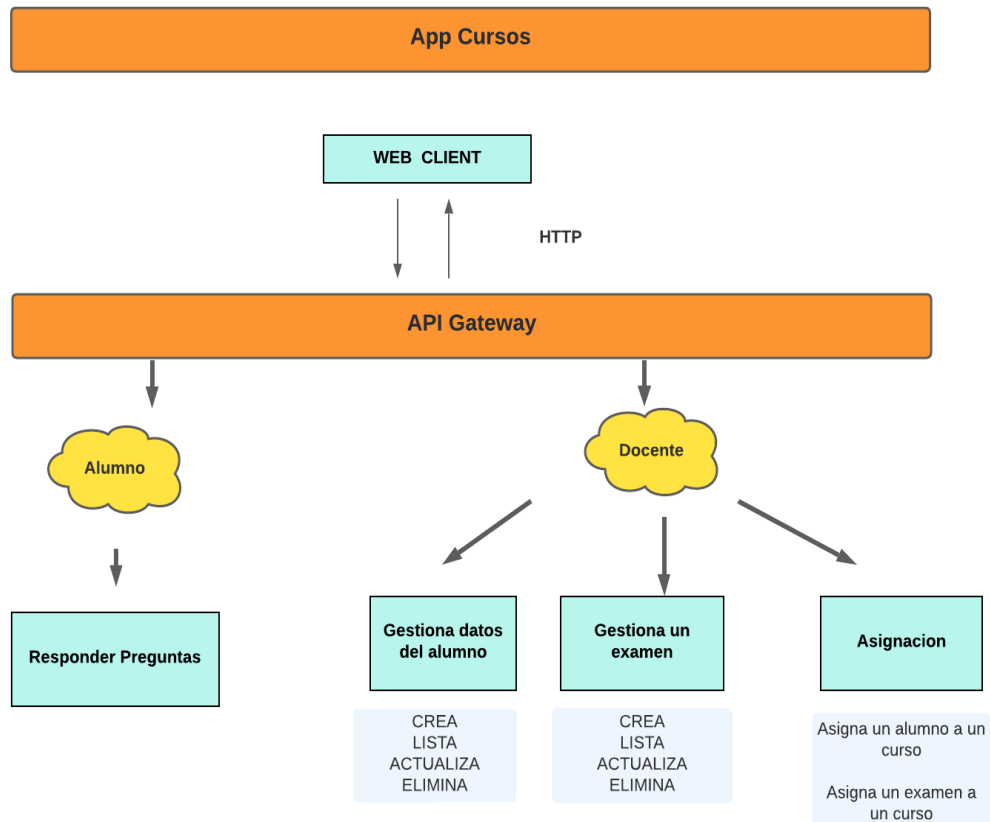


Gráfico 1. Ideograma del Prototipo de la Aplicación Web Cursos

6. FECHAS CRÍTICAS

Fecha inicio: 18 de Abril del 2022

Fecha de entregas de sprint 1: 18 de Mayo

Fecha de entrega de sprint 2: 18 de Junio

Fecha de entrega de sprint 3: 18 de Julio

Fecha de entregas de reléase: 01 de Agosto

Fecha fin 4 de Agosto del 2022

7. MIEMBROS DEL EQUIPO CON ROLES

En la Tabla 1, se muestra los participantes que cumplirán con cada rol y la descripción acorde a las funciones a cumplir.

Tabla 1. Roles de los miembros del Equipo

Rol	Nombre	Actividad
Product Owner	Ing. Verónica Tapia	Representa a las partes interesadas tanto internas como externas, por lo que debe comprender y respaldar las necesidades de todos los usuarios comerciales, así como las necesidades y los procesos del equipo Scrum
Scrum Master	Ing. Verónica Tapia	Su misión es que todo el equipo entienda y aplique scrum correctamente. Su trabajo es que los equipos trabajen hacia sus objetivos hasta llegar a la etapa de "enemigo final", eliminando las dificultades que puedan encontrar en el camino.
Team	Alexandra Salazar, María José Toasa	Son las personas responsables de que el equipo aporte valor al negocio. Representan a las partes interesadas internas y externas, por lo que debe comprender y respaldar las necesidades de todos los usuarios comerciales.

8. PRODUCT BACKLOG (PRIORIZADO)

En la Tabla 2, se muestra la priorización del Product Backlog con 19 Historias de Usuario identificadas de la siguiente manera: 4 Historias de Usuario con prioridad Alta, 7 Historias de Usuario con prioridad Media y 8 Historias de Usuario con prioridad Baja.

8.1. Priorizar Product Backlog

Tabla 2. Priorización del Product Backlog

Identificador de la Historia de Usuario	Descripción de Historias de Usuario	Prioridades
HU001	Como docente requiero registrar un alumno para ser asignado a un curso	Baja
HU002	Como docente requiero mostrar los datos del alumno para obtener una lista de alumnos existentes	Baja
HU003	Como docente requiero modificar los datos del alumno para tener la información más íntegra posible	Baja
HU004	Como docente requiero dar de baja a un alumno para un mejor control de datos	Baja
HU005	Como docente requiero dar de baja a un alumno asignado a un curso para un mejor control de datos	Baja
HU006	Como docente requiero crear un curso para asignar exámenes y alumnos	Alta
HU007	Como docente requiero mostrar un curso para obtener un listado	Alta
HU008	Como docente requiero modificar un curso para tener la información más íntegra posible	Alta
HU009	Como docente requiero dar de baja un curso para un mejor control de datos	Alta
HU010	Como docente requiero asignar un alumno a un curso para que tenga acceso a sus exámenes	Media
HU011	Como docente requiero asignar un examen a un curso para que los alumnos puedan responder	Media
HU012	Como docente requiero crear un examen para asignar a los cursos creados	Media
HU013	Como docente requiero mostrar la lista de exámenes para asignar a un curso específico	Media
HU014	Como docente requiero modificar un examen para tener la información más íntegra posible	Media
HU015	Como docente requiero dar de baja un examen para un mejor control de datos	Media
HU016	Como docente requiero dar de baja un examen asignado a un curso para un mejor control de datos	Media
HU017	Como alumno requiero ver los exámenes respondidos para observar las respuestas	Baja

HU018	Como alumno requiero responder el examen para que se guarde y se pueda visualizar las respuestas	Baja
HU019	Como alumno deseo ver los exámenes asignados para poder responder	Baja

Definir prioridades
Alta
Media
Baja

8.2. Sprint Backlog

Para el Sprint Backlog se identificó 6 historias de usuario con prioridad alta que corresponde al sprint uno, 6 historias de usuario con prioridad media que corresponde al segundo sprint y 6 historias de usuario con prioridad baja que corresponde al tercer sprint, como se muestra en la Tabla 3.

Tabla 3. Priorización del Product Backlog

Orden de Historia de Usuario Designada	Descripción de Historias de Usuario	Prioridad	Sprint	Puntos de historia
HU006	Como docente requiero crear un curso para asignar exámenes y alumnos	Alta	1	8
HU007	Como docente requiero mostrar un curso para obtener un listado	Alta	1	8
HU008	Como docente requiero modificar un curso para tener la información más integra posible	Alta	1	8
HU009	Como docente requiero dar de baja un curso para un mejor control de datos	Alta	1	8
HU012	Como docente requiero crear un examen para asignar a los cursos creados	Media	1	5
HU013	Como docente requiero mostrar la lista de exámenes para asignar a un curso específico	Media	1	5
HU014	Como docente requiero modificar un examen para tener la	Media	2	5

	información más íntegra posible			
HU015	Como docente requiero dar de baja un examen para un mejor control de datos	Media	2	5
HU016	Como docente requiero dar de baja un examen asignado a un curso para un mejor control de datos	Media	2	5
HU001	Como docente requiero registrar un alumno para ser asignado a un curso	Baja	2	3
HU002	Como docente requiero mostrar los datos del alumno para obtener una lista de alumnos existentes	Baja	2	3
HU003	Como docente requiero modificar los datos del alumno para tener la información más íntegra posible	Baja	2	3
HU004	Como docente requiero dar de baja a un alumno para un mejor control de datos	Baja	3	3
HU005	Como docente requiero dar de baja a un alumno asignado a un curso para un mejor control de datos	Baja	3	3
HU010	Como docente requiero asignar un alumno a un curso para que tenga acceso a sus exámenes	Baja	3	3
HU011	Como docente requiero asignar un examen a un curso para que los alumnos puedan responder	Baja	3	3
HU017	Como alumno requiero ver los exámenes respondidos para observar las respuestas	Baja	3	3
HU019	Como alumno deseo ver los exámenes asignados para poder responder	Baja	3	3
HU018	Como alumno requiero responder el examen para que se guarde y se pueda visualizar las respuestas	Baja	3	3
Total Puntos de Historia			87	

9. PRÁCTICAS ÁGILES A USAR

- Historias de usuario
- Product backlog
- Roles
- Release Planning
- Sprint
- Sprint Backlog
- Planning poker

10. TECNOLOGÍA A USAR

En la Tabla 4, se puede observar las herramientas con la versión que se utilizaron para realizar todos los procesos necesarios, de esta manera la web funcione de forma correcta.

Tabla 4. Herramientas Back End

Back End	
Herramienta	Versión
JDK(Java SE Development Kit)	12.0.2
Spring Tools IDE(Eclipse)	4
Maven	
MariaDB(Base de datos relacional)	10.6
PostgreSQL (Base de datos relacional)	12
MongoDB (Base de datos no relacional)	5.0.9
Postman	

En la Tabla 5, se puede observar las herramientas con la versión que se utilizaron para realizar el diseño del prototipo web.

Tabla 5. Herramientas Front End

FRONT END	
Herramienta	Versión
Angular CLI	14.0.5
Package Manager	npm 8.5.5 @angular-devkit/architect 0.1400.5 (cli-only) @angular-devkit/core 14.0.5 (cli-only) @angular-devkit/schematics 14.0.5 (cli-only) @schematics/angular 14.0.5 (cli-only)
Bootstrap	4.6.1
Node	16.15.0
Visual Studio Code	1.69.1

11. MODELO DE LA BASE DE DATOS

Para el diseño de la base de datos del microservicio exámenes se realizó el modelo entidad relación (MER) que consta de una sola entidad con sus atributos, como se observa en la Figura 2.

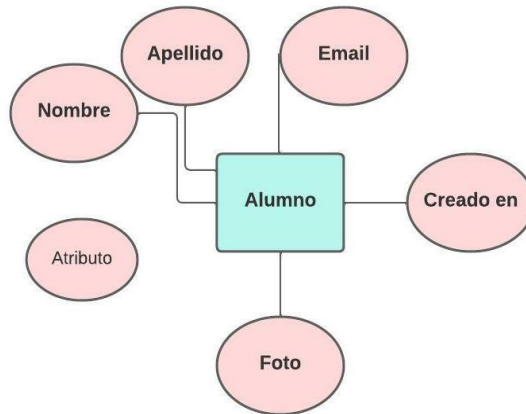


Figura 2. Modelo Entidad Relación del microservicio usuarios

Para el diseño de la base de datos del microservicio exámenes se realizó el modelo entidad relación (MER) que consta de cuatro entidades con sus respectivos atributos, como se observa en la Figura 3.

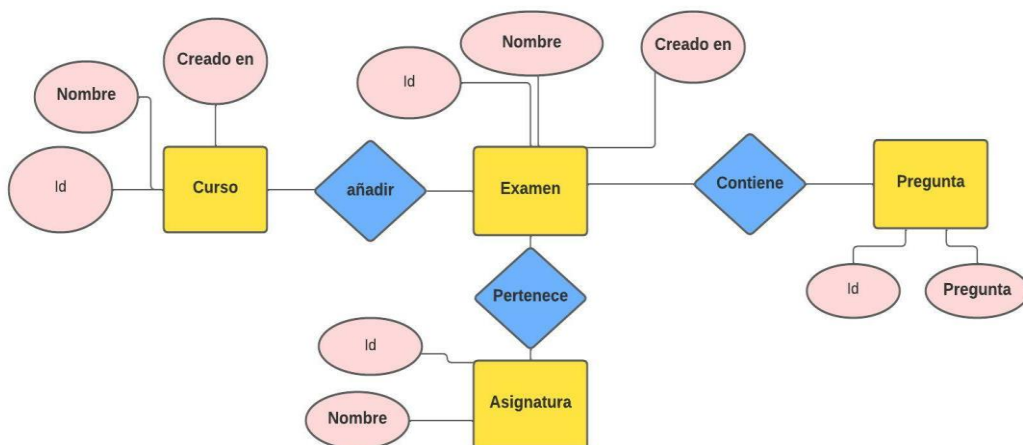


Figura 3. Modelo Entidad Relación del microservicio exámenes

Para el diseño de la base de datos del microservicio respuestas se realizó el modelo de base de datos a través de un grafo que tiene 5 vértices y 4 arcos, como se observa en la Imagen 2.

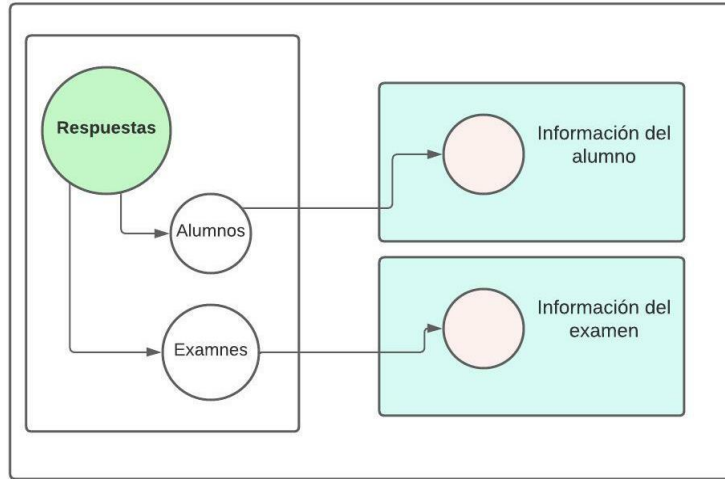


Figura 4. Modelo de base de datos del microservicio respuestas

12. DISEÑO DE LA ARQUITECTURA DE MICROSERVICIOS UTILIZANDO EL MODELO DE OPTIMIZACIÓN

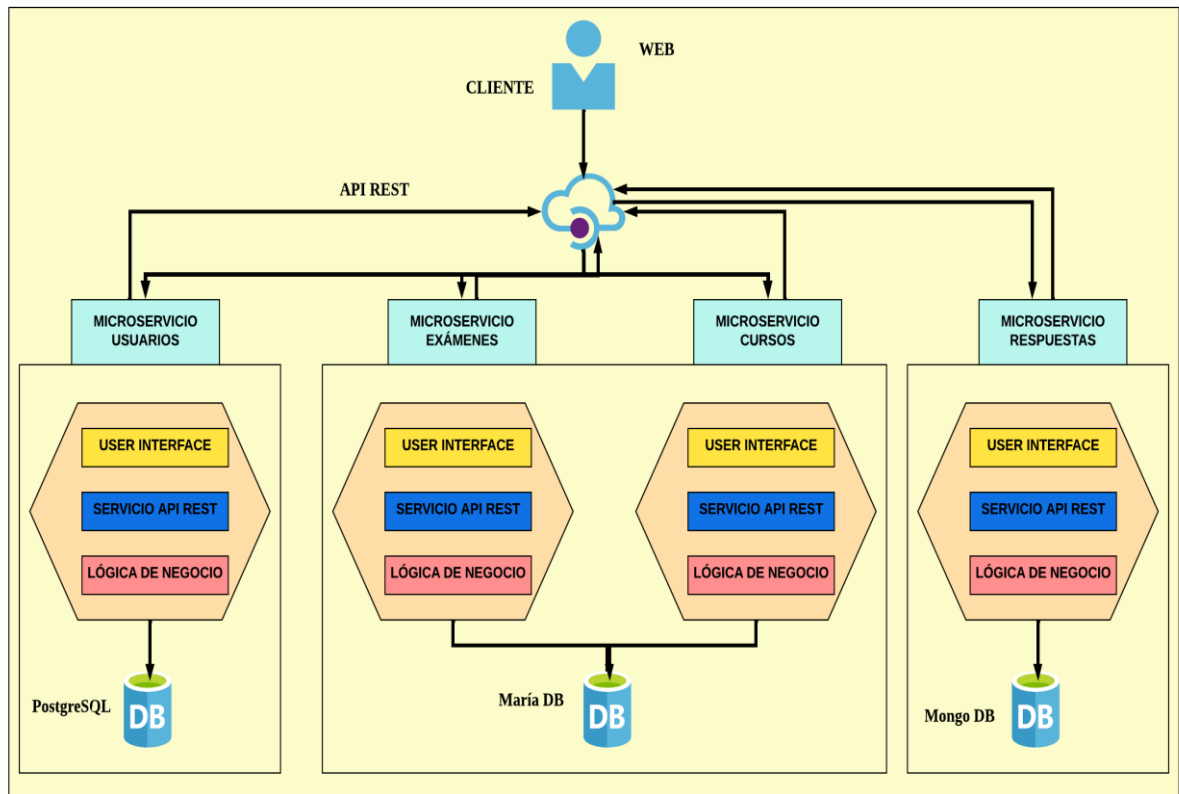


Figura 9. Modelo del diseño de la aplicación

13. RELEASE PLANNING

En la Figura 5, se puede observar el Release Planning que consta de 3 sprint cada uno con sus respectivas Historias de Usuario, estas son las funcionalidades que se van a entregar.

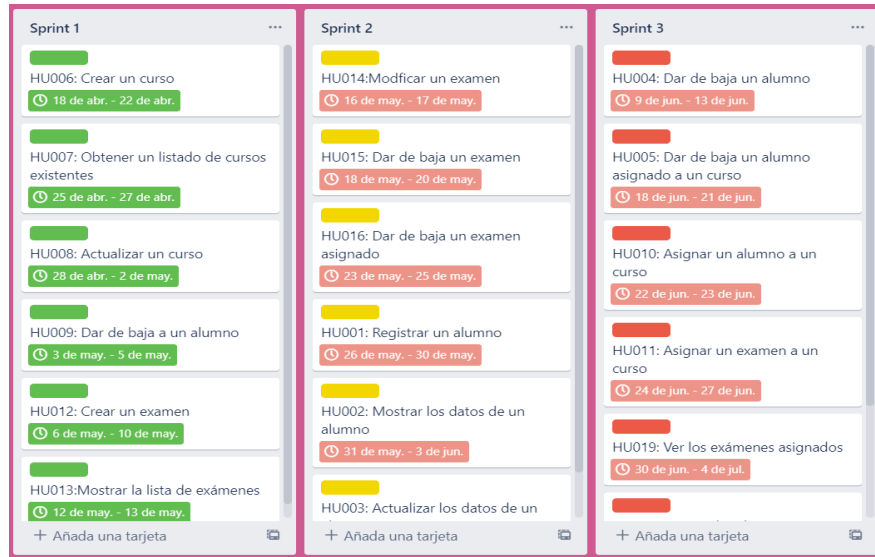


Figura 5. Release Planning

14. PLANNING POKER

Para realizar la estimación de puntos de historias se utilizó la herramienta planIT poker online, la medida que se utilizará es la sucesión de fibonacci según cada Historia Usuario utilizando los valores 3, 5 y 8 que corresponden al tiempo que se va a emplear en el desarrollo por persona en cada día de trabajo.

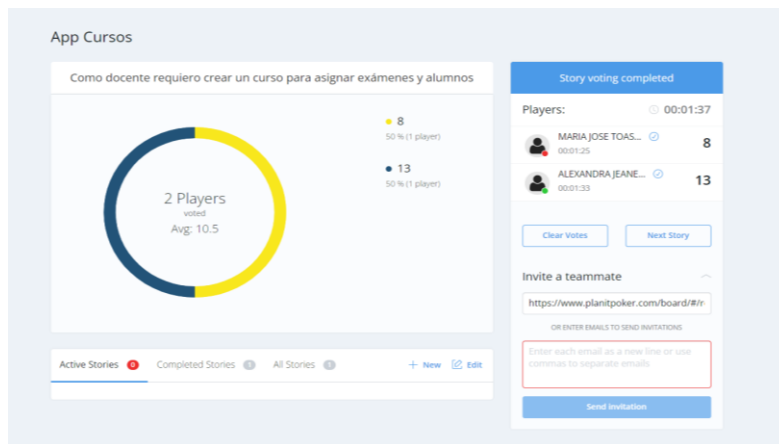


Figura 6. Estimación puntos de Historia

15. DETALLES DE LAS HISTORIAS DE USUARIO

En las tablas 6 a la 24 se detallan las historias de usuario por sprint según la priorización del Product Backlog que se utilizará en cada microservicio.

14.1. Sprint 1

En el siguiente apartado se detallan las historias de usuario que se va a desarrollar en el primer sprint.

Tabla 6. Historia de Usuario HU006 - Crear curso

Historia de Usuario (HU)			
Código HU:	HU006	Fecha:	18/04/2022
Sprint:	1	Prioridad:	Alta
Actor(es):	Docente	Puntos:	8
Descripción:			
Como docente requiero crear un curso para asignar exámenes y alumnos			
Detalle de la HU:			
El Formulario de creación de curso debe contener el siguiente campo: Nombre del Curso			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden acceder a crear un curso.			
Criterios de aceptación:			
1. Dado el formulario de crear un curso cuando ingresa el nombre del curso entonces la aplicación creará un curso mostrando el mensaje “Curso ... creado con éxito” 2. Dado el formulario de crear un curso cuando valide los campos vacíos entonces la aplicación presentará al usuario el mensaje de error “El campo nombre no debe estar vacío”			
DoD(Definition Of Done):			
La historia debe estar verificada y validada			

Tabla 7. Historia de Usuario HU007 - Mostrar curso

Historia de Usuario (HU)			
Código HU:	HU07	Fecha:	18/05/2022
Sprint:	1	Prioridad:	Alta
Actor(es):	Docente	Puntos:	8
Descripción:			
Como docente requiero mostrar un curso para obtener un listado			
Detalle de la HU:			
El Formulario de mostrar curso debe contener el siguiente campo: Nombre del Curso			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden acceder a consultar un curso.			
Criterios de aceptación:			
1. Dado el formulario de consulta de un curso, cuando el docente haga clic en cursos, entonces la aplicación mostrará la lista de cursos creados			
DoD(Definition Of Done):			
1. La historia debe estar verificada y validada			

Tabla 8. Historia de Usuario HU008 - Modificar curso

Historia de Usuario (HU)			
Código HU:	HU008	Fecha:	18/04/2022
Sprint:	1	Prioridad:	Alta
Actor(es):	Docente	Puntos:	8
Descripción:			
Como docente requiero modificar un curso para tener la información más íntegra posible			
Detalle de la HU:			
El Formulario de modificar un curso debe contener el siguiente campo: Nombre del curso			

Restricciones:
1. Solo los usuarios con privilegios de docente pueden acceder a modificar el nombre del curso.
Criterios de aceptación:
<ol style="list-style-type: none"> 1. Dado el formulario que muestre la opción para modificar un curso cuando se haya modificado el campo del Nombre del Curso entonces la aplicación mostrará el mensaje “Curso ... actualizado con éxito” 2. Dado el formulario que muestre la opción para modificar un curso cuando valide los campos vacíos entonces la aplicación presentará al usuario el mensaje de error “El campo nombre no debe estar vacío”
DoD(Definition Of Done):
La historia debe estar verificada y validada

Tabla 9. Historia de Usuario HU009 - Dar de baja curso

Historia de Usuario (HU)			
Código HU:	HU009	Fecha:	18/05/2022
Sprint:	1	Prioridad:	Alta
Actor(es):	Docente	Puntos:	8
Descripción:			
Como docente requiero dar de baja un curso para un mejor control de datos			
Detalle de la HU:			
El Formulario dar de baja un curso debe contener el siguiente campo: Nombre del curso			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden acceder a dar de baja un curso.			
Criterios de aceptación:			
<ol style="list-style-type: none"> 1. Dado el formulario curso que muestre la opción para dar de baja un curso cuando el docente seleccione un curso entonces la aplicación emite un mensaje de “Curso Octavo B dado de baja con éxito” 2. Dado el formulario curso que muestre la opción para dar de baja un curso, cuando el docente seleccione un curso, entonces la aplicación mostrará un mensaje de confirmación “¿Seguro que desea dar de baja a Octavo B?” 			

DoD(Definition Of Done):
1. La historia debe estar verificada y validada

Tabla 10. Historia de Usuario HU012 - Crear examen

Historia de Usuario (HU)			
Código HU:	HU012	Fecha:	18/05/2022
Sprint:	1	Prioridad:	Media
Actor(es):	Docente	Puntos:	5
Descripción:			
Como docente requiero crear un examen para asignar a los cursos creados			
Detalle de la HU:			
El Formulario de creación de exámenes debe contener los siguientes campos: Nombre del examen, Asignatura, Tema, Preguntas			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden crear un examen			
Criterios de aceptación:			
<ol style="list-style-type: none"> 1. Dado el formulario de crear un examen cuando el usuario registre los datos Nombre del examen, Asignatura, Tema, Pregunta correctamente la aplicación emite el mensaje “Examen.... creado con éxito”. 2. Dado el formulario de crear un examen cuando el usuario registre el Tema que exceda en un rango de 3 a 40 caracteres la aplicación emite el mensaje de error “El tamaño del campo está entre 4 y 30 caracteres”. 3. Dado el formulario de crear un examen cuando valida los campos y se encuentran vacíos la aplicación informará al usuario de los posibles errores del formulario 			
DoD(Definition Of Done):			
La historia debe estar verificada y validada			

Tabla 11. Historia de Usuario HU013 - Mostrar exámenes

Historia de Usuario (HU)			
Código HU:	HU013	Fecha:	18/05/2022
Sprint:	1	Prioridad:	Media
Actor(es):	Docente	Puntos:	5
Descripción:			
Como docente requiero mostrar la lista de exámenes para asignar a un curso específico			
Detalle de la HU:			
El Formulario de mostrar la lista de exámenes debe contener los siguientes campos: Nombre del examen, Asignatura, Tema, Preguntas			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden acceder a mostrar la lista de exámenes.			
Criterios de aceptación:			
1. Dado el formulario de exámenes, cuando el docente haga clic en exámenes, entonces la aplicación mostrará la lista de exámenes creados			
DoD(Definition Of Done):			
La historia debe estar verificada y validada			

14.2. Sprint 2

En el siguiente apartado se detallan las historias de usuario que se va a desarrollar en el segundo sprint.

Tabla 12. Historia de Usuario HU014 - Modificar examen

Historia de Usuario (HU)			
Código HU:	HU014	Fecha:	18/06/2022
Sprint:	2	Prioridad:	Media
Actor(es):	Docente	Puntos:	5

Descripción:
Como docente requiero modificar un examen para tener la información más íntegra posible
Detalle de la HU:
El Formulario de modificar exámenes debe contener los siguientes campos: Nombre del examen, Asignatura, Tema, Preguntas
Restricciones:
1. Solo los usuarios con privilegios de docente pueden modificar un examen.
Criterios de aceptación:
<ol style="list-style-type: none"> 1. Dado el formulario de examen que muestre la opción para modificar la información del examen cuando ingresa los datos de forma correcta entonces la aplicación mostrará el siguiente mensaje “Examen Examen de Historia actualizado con éxito” 2. Dado el formulario de examen que muestre la opción para modificar la información del examen cuando el campo nombre sea menor que 4 caracteres y mayor que 30 caracteres entonces la aplicación presentará al usuario el mensaje de error “El campo nombre el tamaño debe estar entre 4 y 30” 3. Dado el formulario de examen que muestre la opción para modificar la información del examen cuando el examen no contiene preguntas entonces la aplicación presentará al usuario el mensaje de error “Examen debe tener preguntas”
DoD(Definition Of Done):
La historia debe estar verificada y validada

Tabla 13. Historia de Usuario HU015 - Dar de baja examen

Historia de Usuario (HU)			
Código HU:	HU015	Fecha:	18/06/2022
Sprint:	2	Prioridad:	Media
Actor(es):	Docente	Puntos:	5
Descripción:			
Como docente requiero dar de baja un examen para un mejor control de datos			
Detalle de la HU:			

El Formulario dar de baja exámenes debe contener los siguientes campos: Nombre del examen, Asignatura, Tema, Preguntas
Restricciones:
1. Solo los usuarios con privilegios de docente pueden acceder a dar de baja un examen.
Criterios de aceptación:
<ol style="list-style-type: none"> 1. Dado el formulario examen que muestre la opción para dar de baja un examen, cuando el docente seleccione un examen, entonces la aplicación emite un mensaje “Examen Gracia dado de baja con éxito” 2. Dado el formulario examen que muestre la opción para dar de baja un examen, cuando el docente seleccione un examen, entonces la aplicación mostrará un mensaje de confirmación “¿Seguro que desea dar de baja a Gracia?”
DoD(Definition Of Done):
La historia debe estar verificada y validada

Tabla 14. Historia de Usuario HU016 - Dar de baja examen asignado

Historia de Usuario (HU)			
Código HU:	HU016	Fecha:	18/06/2022
Sprint:	2	Prioridad:	Media
Actor(es):	Docente	Puntos:	5
Descripción:			
Como docente requiero dar de baja un examen asignado a un curso para un mejor control de datos			
Detalle de la HU:			
El Formulario para dar de baja un examen asignado debe mostrar la opción Dar de baja.			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden acceder a dar de baja un examen asignado a un curso			
Criterios de aceptación:			
1. Dado el formulario curso cuando el examen se encuentra asignado a un curso entonces la aplicación mostrará todos los exámenes asignados y emite un mensaje de “ Eliminación exitosa”			

2. Dado el formulario curso cuando el examen se encuentra asignado a un curso entonces la aplicación mostrará todos los exámenes asignados y emite un mensaje de confirmación “ ¿Seguro que desea dar de baja a ... ?”
DoD(Definition Of Done):
La historia debe estar verificada y validada

Tabla 15. Historia de Usuario HU001 - Registrar alumno

Historia de Usuario (HU)			
Código HU:	HU001	Fecha:	18/06/2022
Sprint:	2	Prioridad:	Baja
Actor(es):	Docente	Puntos:	3
Descripción:			
Como docente requiero registrar un alumno para ser asignado a un curso			
Detalle de la HU:			
El Formulario de registro de alumno debe contener los siguientes campos: Nombre, Apellido, Email, Foto			
Restricciones:			
<ol style="list-style-type: none"> Solo los usuarios con privilegios de docente pueden acceder a registrar los datos del alumno Se debe completar todos los campos 			
Criterios de aceptación:			
<ol style="list-style-type: none"> Dado el formulario de registro de alumnos cuando ingresa los datos de forma correcta entonces la aplicación creará un nuevo alumno, mostrará el siguiente mensaje “Alumno.... creado con éxito” Dado el formulario de registro de alumnos cuando cargue la fotografía con un formato incorrecto entonces la aplicación emite el mensaje “El archivo debe ser del tipo imagen” Dado el formulario de registro de alumnos cuando valide los campos vacíos entonces la aplicación presentará al usuario el mensaje de error “El campo no debe estar vacío” 			
DoD (Definition Of Done):			
La historia debe estar verificada y validada			

Tabla 16. Historia de Usuario HU002 - Mostrar alumno

Historia de Usuario (HU)			
Código HU:	HU002	Fecha:	18/06/2022
Sprint:	2	Prioridad:	Baja
Actor(es):	Docente	Puntos:	3
Descripción:			
Como docente requiero mostrar los datos del alumno para obtener una lista de alumnos existentes			
Detalle de la HU:			
Nombre, apellido, teléfono, foto.			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden acceder a mostrar los datos del alumno.			
Criterios de aceptación:			
1. Dado el formulario de alumnos, cuando el docente haga clic en alumnos, entonces la aplicación mostrará la lista de alumnos creados			
DoD(Definition Of Done):			
La historia debe estar verificada y validada			

Tabla 17. Historia de Usuario HU003 - Modificar alumno

Historia de Usuario (HU)			
Código HU:	HU003	Fecha:	18/06/2022
Sprint:	2	Prioridad:	Baja
Actor(es):	Docente	Puntos:	3
Descripción:			
Como docente requiero modificar los datos del alumno para tener la información más íntegra posible			
Detalle de la HU:			

El Formulario de actualización de datos debe contener los siguientes campos: Nombre, Apellido, Email, Foto
Restricciones:
1. Solo los usuarios con privilegios de docente pueden acceder a modificar los datos de un alumno
Criterios de aceptación:
<ol style="list-style-type: none"> 1. Dado el formulario que muestre la opción para modificar la información del alumno cuando ingresa los datos de forma correcta entonces la aplicación mostrará el siguiente mensaje “Alumno actualizado con éxito” 2. Dado el formulario que muestre la opción para modificar la información del alumno cuando cargue la fotografía con un formato incorrecto entonces la aplicación emite el mensaje “El archivo debe ser del tipo imagen” 3. Dado el formulario que muestre la opción para modificar la información del alumno cuando valide los campos vacíos entonces la aplicación presentará al usuario el mensaje de error “El campo no debe estar vacío”
DoD(Definition Of Done):
La historia debe estar verificada y validada

14.3. Sprint 3

En el siguiente apartado se detallan las historias de usuario que se va a desarrollar en el primer sprint.

Tabla 18. Historia de Usuario HU004 - Dar de baja alumno

Historia de Usuario (HU)			
Código HU:	HU004	Fecha:	18/07/2022
Sprint:	3	Prioridad:	Baja
Actor(es):	Docente	Puntos:	3
Descripción:			
Como docente requiero dar de baja a un alumno para un mejor control de datos			
Detalle de la HU:			
El Formulario despliega la información de los alumnos con una opción para dar de baja al alumno			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden acceder a dar de baja a un alumno.			

Criterios de aceptación:
<ol style="list-style-type: none"> 1. Dado el formulario que muestre los alumnos con la opción para dar de baja cuando selecciona al alumno entonces la aplicación mostrará un mensaje de “ Eliminación exitosa” 2. Dado el formulario que muestre los alumnos con la opción para dar de baja cuando selecciona al alumno entonces la aplicación mostrará un mensaje de confirmación “ ¿Seguro que desea dar de baja al Alumno . ?”
DoD(Definition Of Done):
La historia debe estar verificada y validada

Tabla 19. Historia de Usuario HU005 - Dar de baja alumno asignado

Historia de Usuario (HU)			
Código HU:	HU005	Fecha:	18/07/2022
Sprint:	3	Prioridad:	Baja
Actor(es):	Docente	Puntos:	3
Descripción:			
Como docente requiero dar de baja a un alumno asignado a un curso para un mejor control de datos			
Detalle de la HU:			
Nombre, apellido, teléfono, foto.			
Restricciones:			
<ol style="list-style-type: none"> 1. Solo los usuarios con privilegios de docente pueden acceder a dar de baja a un alumno asignado a un curso 			
Criterios de aceptación:			
<ol style="list-style-type: none"> 1. Dado el formulario curso cuando el alumno se encuentra asignado a un curso entonces la aplicación mostrará todos los alumnos asignados y emite un mensaje de “Alumno María José eliminado con éxito del curso Primero B” 2. Dado el formulario curso cuando el alumno se encuentra asignado a un curso entonces la aplicación mostrará todos los alumnos asignados y emite un mensaje de confirmación “ ¿Seguro que desea dar de baja a María José ?” 			
DoD(Definition Of Done):			
La historia debe estar verificada y validada			

Tabla 20. Historia de Usuario HU010 - Asignar un alumno a un curso

Historia de Usuario (HU)			
Código HU:	HU010	Fecha:	18/07/2022
Sprint:	3	Prioridad:	Baja
Actor(es):	Docente	Puntos:	3
Descripción:			
Como docente requiero asignar un alumno a un curso para que tenga acceso a sus exámenes			
Detalle de la HU:			
El formulario del curso permite añadir alumnos.			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden acceder a eliminar un curso.			
Criterios de aceptación:			
1. Dado el formulario cursos para añadir un alumno cuando el curso se encuentra creado entonces la aplicación muestra el curso para asignar alumnos. 2. Dado el formulario cursos para añadir un alumno cuando el alumno ya se encuentra asignado entonces la aplicación muestra un mensaje de error.			
DoD(Definition Of Done):			
La historia debe estar verificada y validada			

Tabla 21. Historia de Usuario HU011 - Asignar un examen a un curso

Historia de Usuario (HU)			
Código HU:	HU011	Fecha:	18/07/2022
Sprint:	3	Prioridad:	Baja
Actor(es):	Docente	Puntos:	3
Descripción:			
Como docente requiero asignar un examen a un curso para que los alumnos puedan responder			
Detalle de la HU:			

El formulario del curso permite añadir exámenes.
Restricciones:
1. Solo los usuarios con privilegios de docente pueden acceder a asignar exámenes a curso.
Criterios de aceptación:
1. Dado el formulario cursos para añadir un examen, cuando el curso se encuentra creado, entonces la aplicación muestra el curso para asignar exámenes. 2. Dado el formulario cursos para añadir un examen cuando el examen ya se encuentra asignado entonces la aplicación muestra un mensaje de error.
DoD(Definition Of Done):
La historia debe estar verificada y validada

Tabla 22. Historia de Usuario HU017 - Ver exámenes respondidos

Historia de Usuario (HU)			
Código HU:	HU017	Fecha:	18/07/2022
Sprint:	3	Prioridad:	Baja
Actor(es):	Docente	Puntos:	3
Descripción:			
Como alumno requiero ver los exámenes respondidos para observar las respuestas			
Detalle de la HU:			
Nombre, creado en y preguntas.			
Restricciones:			
1. Solo los usuarios con privilegios de docente pueden acceder a visualizar los exámenes respondidos.			
Criterios de aceptación:			
1. Dado el formulario alumnos cuando el alumno haya respondido el examen entonces la aplicación le permitirá ver el examen con sus preguntas.			
DoD(Definition Of Done):			
La historia debe estar verificada y validada			

Tabla 23. Historia de Usuario HU019 - Ver exámenes asignados

Historia de Usuario (HU)			
Código HU:	HU019	Fecha:	18/07/2022
Sprint:	3	Prioridad:	Baja
Actor(es):	Alumno	Puntos:	3
Descripción:			
Como alumno deseo ver los exámenes asignados para poder responder			
Detalle de la HU:			
Nombre, creado en y preguntas.			
Restricciones:			
1. Solo los usuarios con privilegios de alumno pueden acceder a ver los exámenes asignados.			
Criterios de aceptación:			
1. Dado el formulario de alumnos, cuando el alumno haga clic en el examen, entonces la aplicación mostrará los exámenes asignados y le permitirá responder. 2. Dado el formulario de alumnos, cuando el alumno haga clic en enviar, entonces la aplicación mostrará el mensaje “Preguntas enviadas con éxito ”			
DoD(Definition Of Done):			
La historia debe estar verificada y validada			

Tabla 24. Historia de Usuario HU018 Responder examen

Historia de Usuario (HU)			
Código HU:	HU18	Fecha:	18/07/2022
Sprint:	3	Prioridad:	Baja
Actor(es):	Alumno	Puntos:	3
Descripción:			
Como alumno requiero responder el examen para que se guarde y se pueda visualizar las respuestas			
Detalle de la HU:			

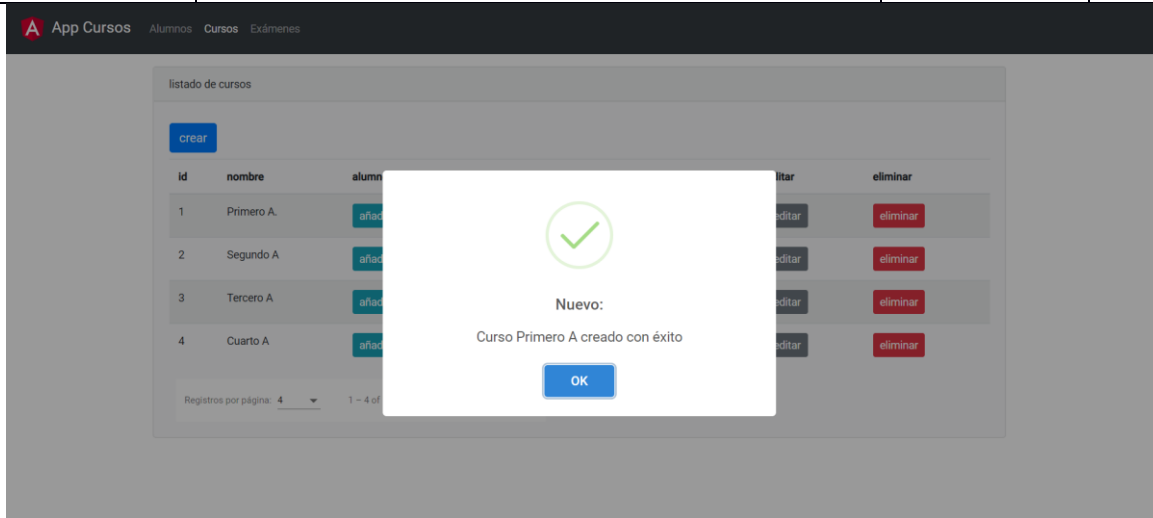
El Formulario de alumno muestra la opción: Respuesta
Restricciones:
1. Solo los usuarios con privilegios de alumno y que estén asignados a un curso pueden responder las preguntas del examen asignado al curso.
Criterios de aceptación:
1. Dado el formulario del alumno cuando esté asignado a un curso entonces la aplicación mostrará la lista de los exámenes asignados al alumno para que responda.
DoD(Definition Of Done):
La historia debe estar verificada y validada

16.Pruebas

15.1. Sprint 1

En las tablas 25 a la 43 se realizaron las pruebas funcionales de cada Historia de Usuario por sprint según la priorización del Product Backlog.

Tabla 25. Casos de Prueba N 006 - Crear curso

N°:	006	Objetivo:	Crear un curso	Usuario:	Docente
Descripción:	Como docente requiero crear un curso para asignar exámenes y alumnos				
Condiciones de entrada:	<ul style="list-style-type: none"> • Estar autenticado como Docente • Abrir el formulario para agregar el nombre del curso 				
Entrada 1:	Nombre: PRIMERO A				
Resultado esperado:	Curso PRIMERO A creado con éxito	Evaluación de la prueba	Si		
					
Entrada 2:	Si existen campos vacíos				
Resultado	Mensaje de error "El campo nombre no debe estar vacío"	Evaluación de	Si		

esperado:		la prueba	
------------------	--	------------------	--

Crear Curso

[volver](#)

Nombre

El campo nombre no debe estar vacío

[Crear](#)

Tabla 26. Casos de Prueba N 007 - Mostrar cursos

N°:	007	Objetivo:	Mostrar cursos	Usuario:	Docente
Descripción:	Como docente requiero mostrar un curso para obtener un listado				
Condiciones de entrada:	<ul style="list-style-type: none"> • Estar autenticado • Abrir el formulario que muestra todos los cursos 				
Entrada 1:	No aplica				
Resultado esperado:	Muestra la lista de todos los cursos creados			Evaluación de la prueba	Si

App Cursos
Alumnos Cursos Exámenes

listado de cursos

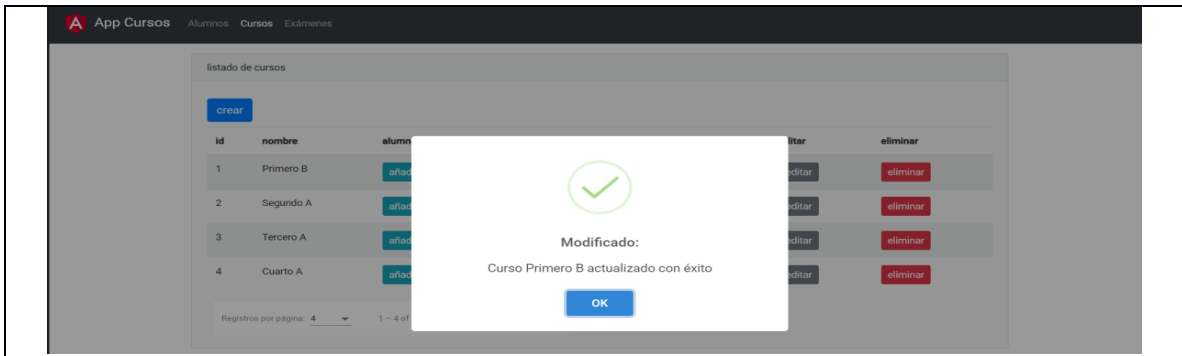
[crear](#)

id	nombre	alumnos	exámenes	creado en	editar	eliminar
1	Primero A.	añadir (5)	añadir (2)	2022/05/21	editar	eliminar
2	Segundo A	añadir (1)	añadir (0)	2022/05/21	editar	eliminar
3	Tercero A	añadir (0)	añadir (1)	2022/05/22	editar	eliminar
4	Cuarto A	añadir (0)	añadir (1)	2022/05/22	editar	eliminar

Registros por página: 4 1 - 4 of 11 |< > >|

Tabla 27. Casos de Prueba N 008 - Modificar cursos

N°:	008	Objetivo:	Modificar un curso	Usuario:	Docente
Descripción:	Como docente requiero modificar el nombre de un curso para tener la información más íntegra posible				
Condiciones de entrada:	<ul style="list-style-type: none"> • Estar autenticado • Abrir el formulario para modificar el nombre del Curso 				
Entrada 1:	Nombre: Primero B,				
Resultado esperado:	Curso Primero B actualizado con éxito			Evaluación de la prueba	Si



Entrada 2:	Si existen campos vacíos		
Resultado esperado:	Mensaje de error por campos requeridos	Evaluación de la prueba	Si

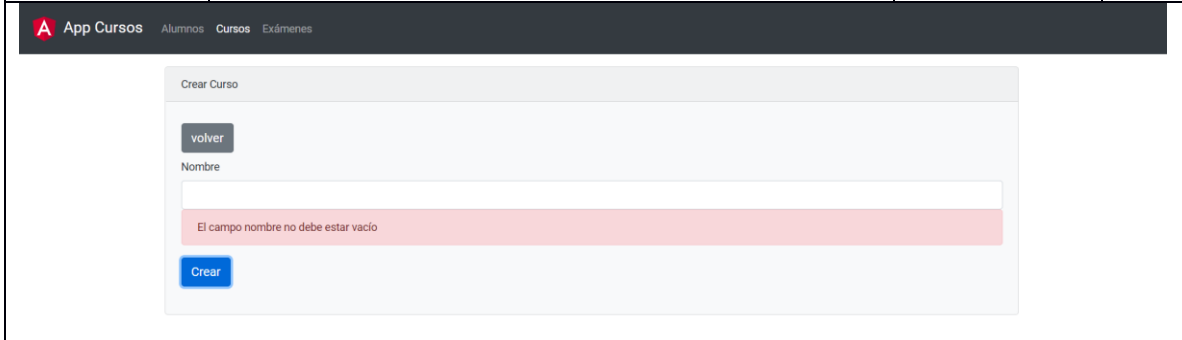
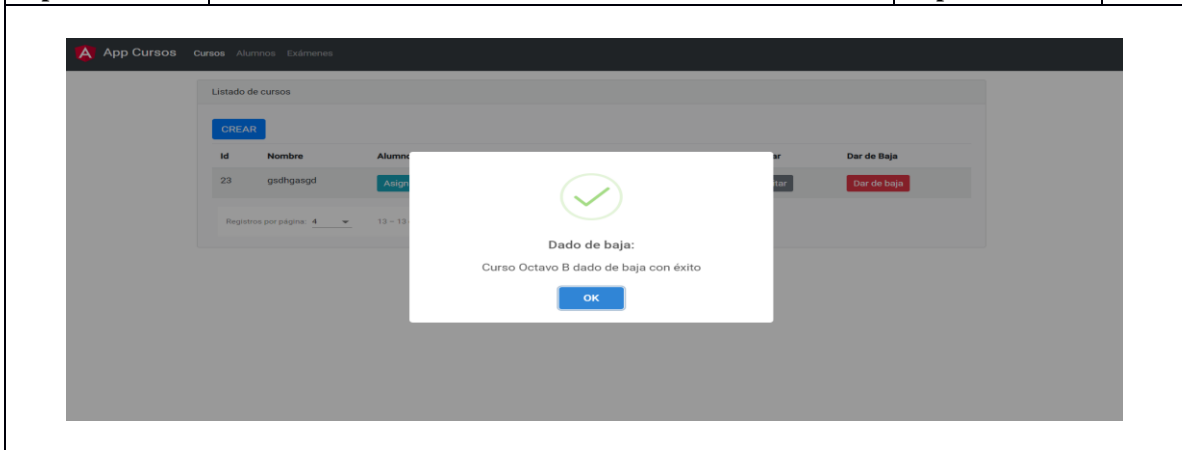


Tabla 28. Casos de Prueba N 009 - Dar de baba curso

N°:	009	Objetivo:	Dar de baja un curso	Usuario:	Docente
Descripción:	Como docente requiero dar de baja un curso para un mejor control de datos				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Estar autenticado ● Abrir el formulario que muestra la lista de cursos ● Seleccionar el curso 				
Entrada 1:	No aplica				
Resultado esperado:	Mensaje de alerta: “Curso Octavo B dado de baja con éxito”	Evaluación de la prueba	Si		



Entrada 2:	No aplica		
Resultado esperado:	Mensaje de alerta “Seguro que desea dar de baja a Octavo B”	Evaluación de la prueba	Si

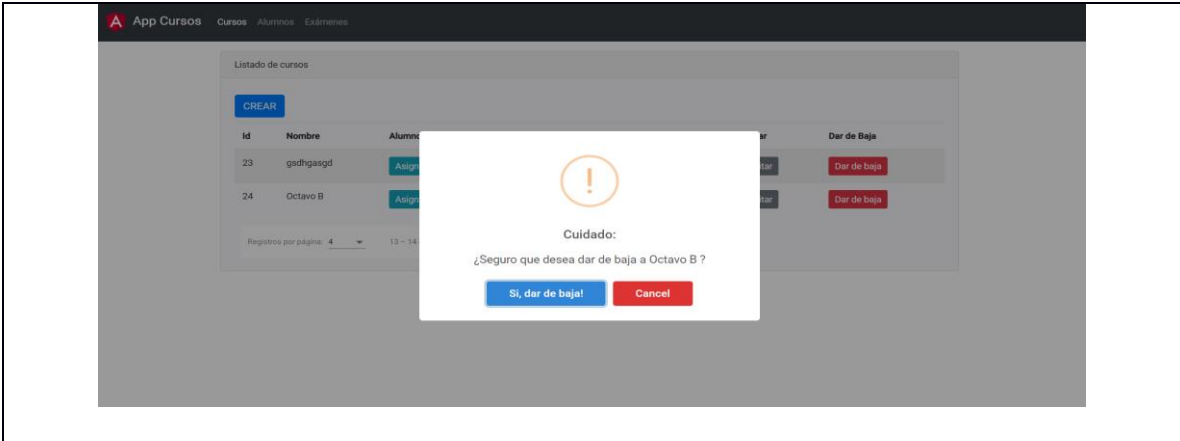
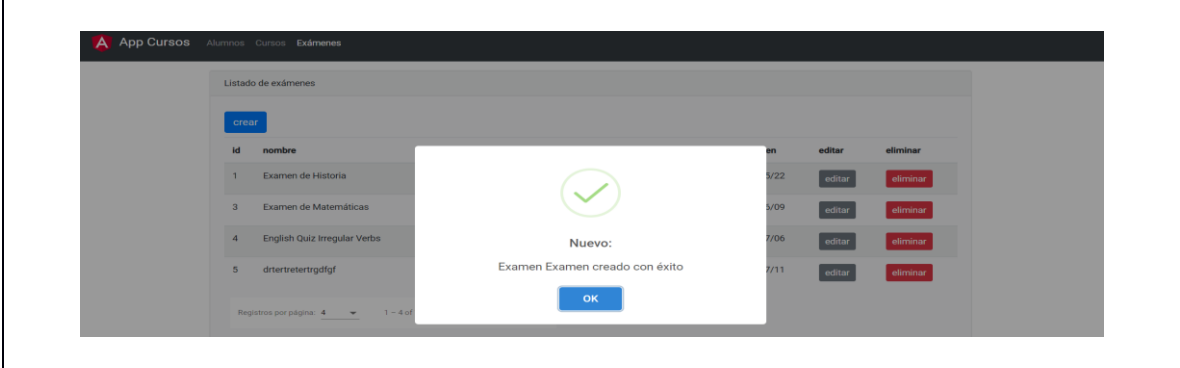
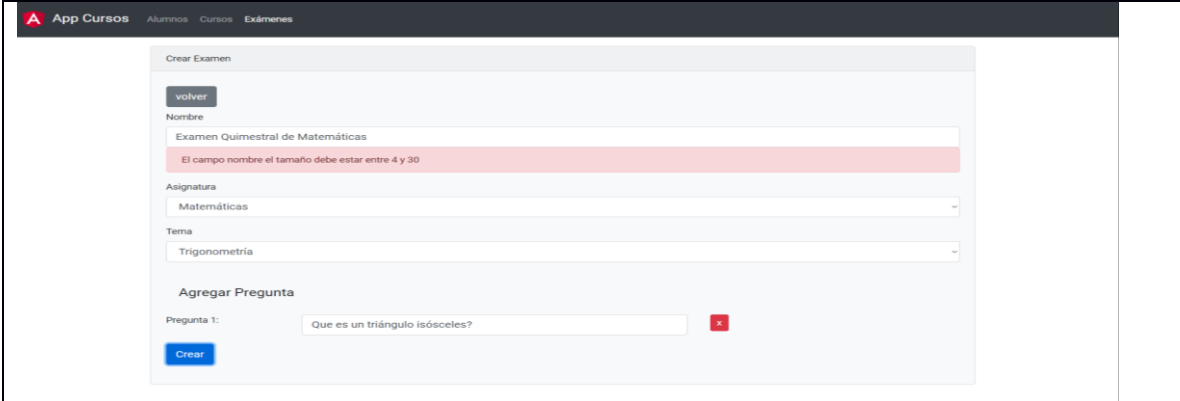


Tabla 29. Casos de Prueba N 012 - Crear examen

Nº:	012	Objetivo:	Crear un Examen	Usuario:	Docente
Descripción:	Como docente requiero crear un examen para asignar a los cursos creados				
Condiciones de entrada:	<ul style="list-style-type: none"> • Estar autenticado • Abrir el formulario para agregar los datos del examen 				
Entrada 1:	Nombre: Matemáticas, Asignatura: Algebra, Tema: Trigonometría Preguntas: ¿Qué es el triángulo isósceles?				
Resultado esperado:	Examen Matemáticas creado con éxito			Evaluación de la prueba	Si



Entrada 2:	Nombre: Evaluación Quimestral de Matemáticas, Nombre: Matemáticas, Asignatura: Algebra,				
Resultado esperado:	Mensaje de error “El campo nombre el tamaño debe estar entre 4 y 30” por campos inválidos			Evaluación de la prueba	Si



Entrada 3:	Si existen campos vacíos		
Resultado esperado:	Mensaje de error por campos requeridos	Evaluación de la prueba	Si

Tabla 30. Casos de Prueba N 013 – Mostrar exámenes

N°:	013	Objetivo:	Mostrar exámenes	Usuario:	Docente
Descripción:	Como docente requiero mostrar la lista de exámenes para asignar a un curso específico				
Condiciones de entrada:	<ul style="list-style-type: none"> • Estar autenticado • Abrir el formulario que muestra todos los cursos 				
Entrada 1:	No aplica				
Resultado esperado:	Muestra la lista de todos los exámenes creados	Evaluación de la prueba	Si		

Id	Nombre	Asignaturas	Preguntas	Fecha	Editar	Dar de Baja
1	Examen de Historia	Ciencias Sociales y Historia -> Ciencias Sociales	3	2022/05/22	Editar	Dar de baja
3	Examen de Matemáticas	Matemáticas -> Trigonometría	3	2022/06/09	Editar	Dar de baja
4	English Quiz Irregular Verbs	Inglés -> Gramática	4	2022/07/06	Editar	Dar de baja
10	Gracia	Lenguaje -> Lectura y comprensión	1	2022/07/29	Editar	Dar de baja

15.2. Sprint 2

Tabla 31. Casos de Prueba N 014 - Modificar examen

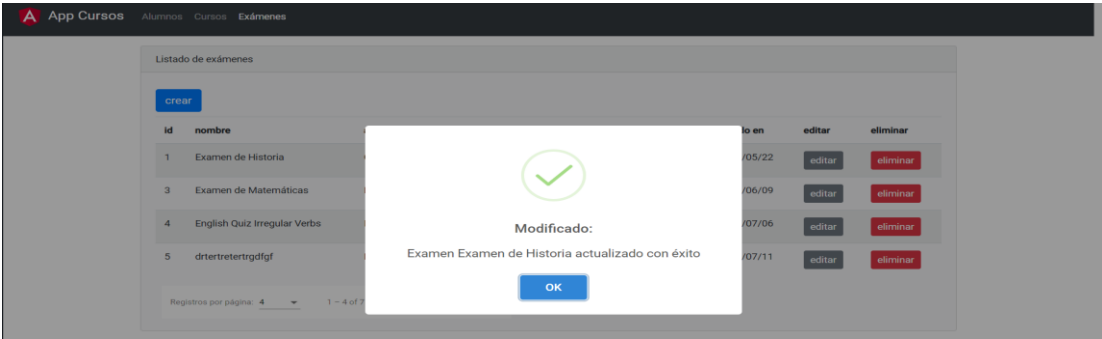
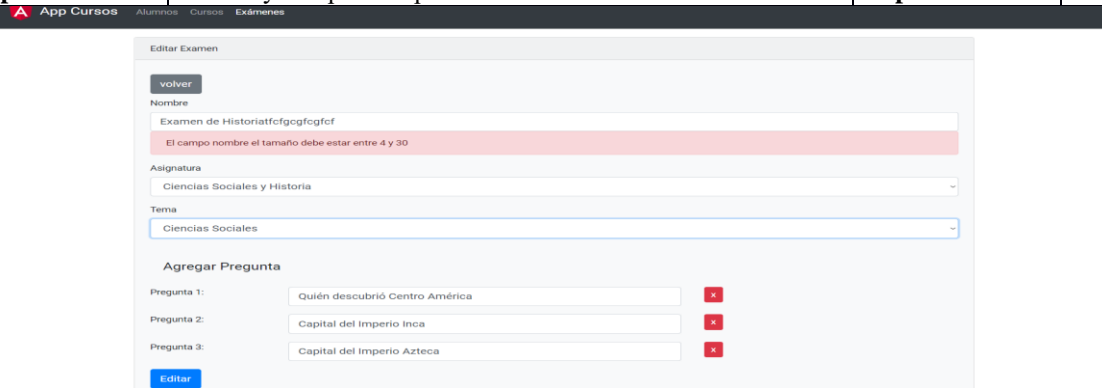
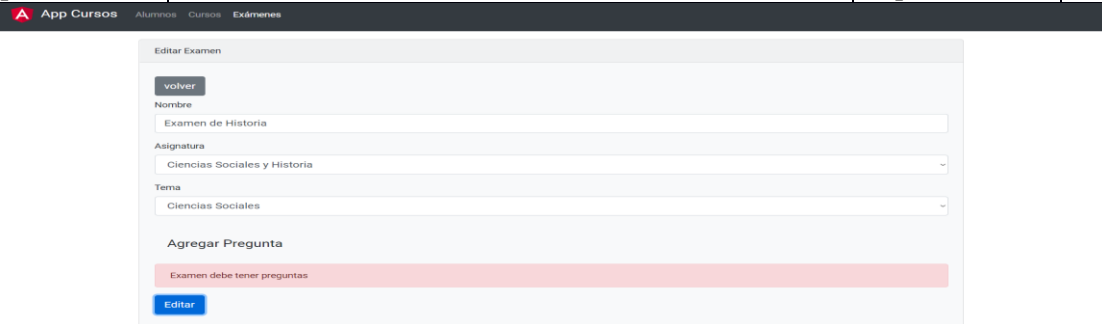
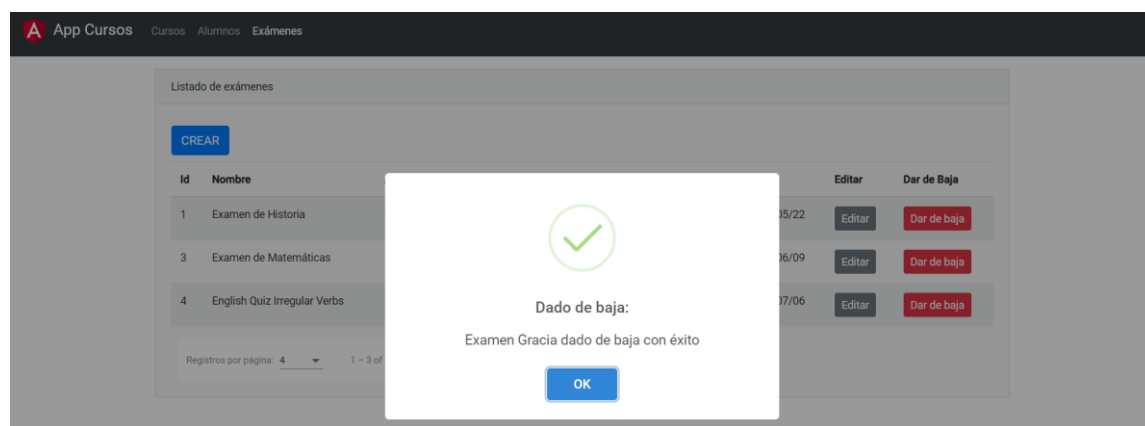
N°:	014	Objetivo:	Modificar un examen	Usuario:	Docente
Descripción:	Como docente requiero modificar un examen para tener la información más integra posible				
Condiciones de entrada:	<ul style="list-style-type: none"> • Estar autenticado • Abrir el formulario para agregar los datos del examen 				
Entrada 1:	Nombre: Examen de Historia, Asignatura: Ciencias Sociales e Historia Asignatura hija: Historia				
Resultado esperado:	Examen Examen de Historia actualizado con éxito	Evaluación de la prueba	Si		
					
Entrada 2:	Nombre: Examen de Historia evaluación quimestral, Asignatura: Ciencias Sociales e Historia Asignatura hija: Historia				
Resultado esperado:	Mensaje de error “El campo nombre el tamaño debe estar entre 4 y 30” por campos inválidos	Evaluación de la prueba	Si		
					
Entrada 3:	Examen sin preguntas				
Resultado esperado:	Mensaje de error “Examen debe tener preguntas”	Evaluación de la prueba	Si		
					

Tabla 32. Casos de Prueba N 015 - Dar de baja examen

N°:	015	Objetivo:	Dar de baja un examen	Usuario:	Docente
Descripción:	Como docente requiero dar de baja un examen para un mejor control de datos				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Estar autenticado ● Abrir el formulario que muestra la lista de exámenes ● Seleccionar el examen 				
Entrada 1:	No aplica				
Resultado esperado:	Mensaje de alerta: “Examen Gracia dado de baja con éxito”	Evaluación de la prueba	Si		



Entrada 2:	No aplica				
Resultado esperado:	Mensaje de alerta “Seguro que desea dar de baja a Gracia”	Evaluación de la prueba	Si		

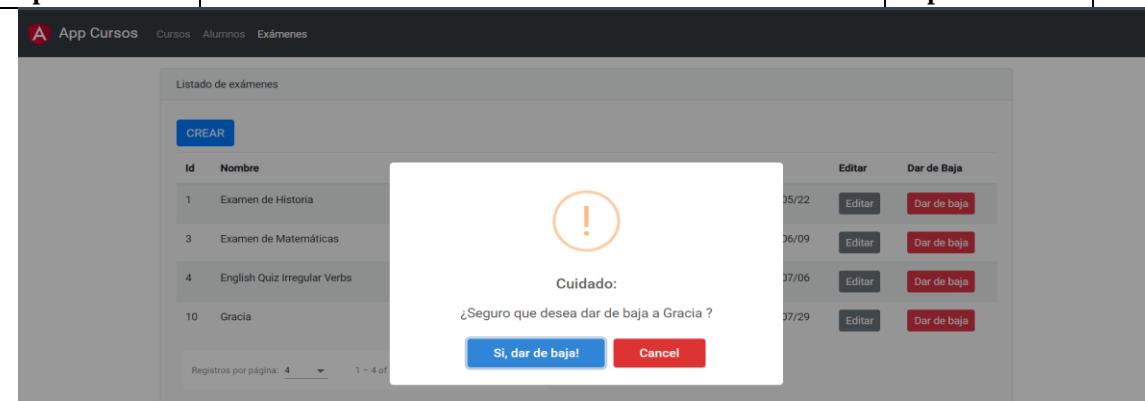
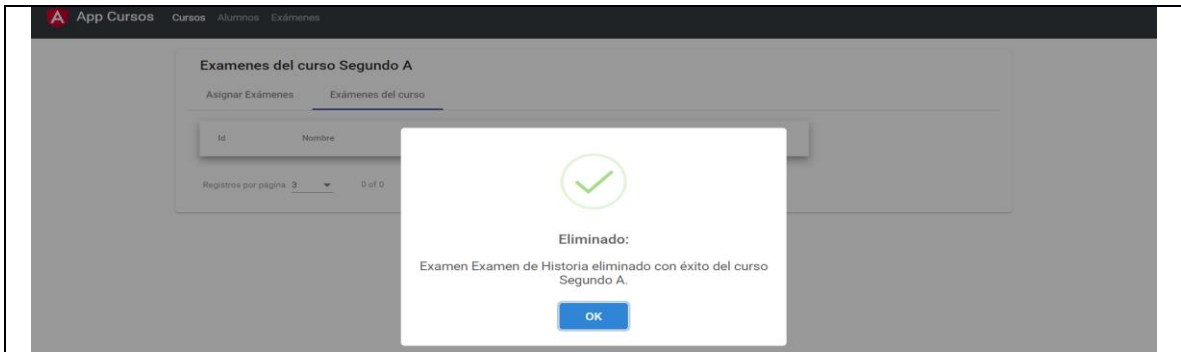


Tabla 33. Casos de Prueba N 016 - Dar de baja un examen asignado

N°:	016	Objetivo:	Dar de baja un examen asignado	Usuario:	Docente
Descripción:	Como docente requiero dar de baja un examen asignado a un curso para un mejor control de datos				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Abrir el formulario de eliminar un examen ● Estar autenticado ● Seleccionar un examen 				
Entrada 1:	No Aplica				
Resultado esperado:	Confirmación de examen para dar de baja	Evaluación de la prueba	Si		



Entrada 2:	No aplica		
Resultado esperado:	Si el docente decide no dar de baja a un examen asignado presiona el botón “Cancelar”, vuelve a la pantalla principal de la sección “Cursos”. Mensaje de alerta “Seguro desea dar de baja”	Evaluación de la prueba	Si

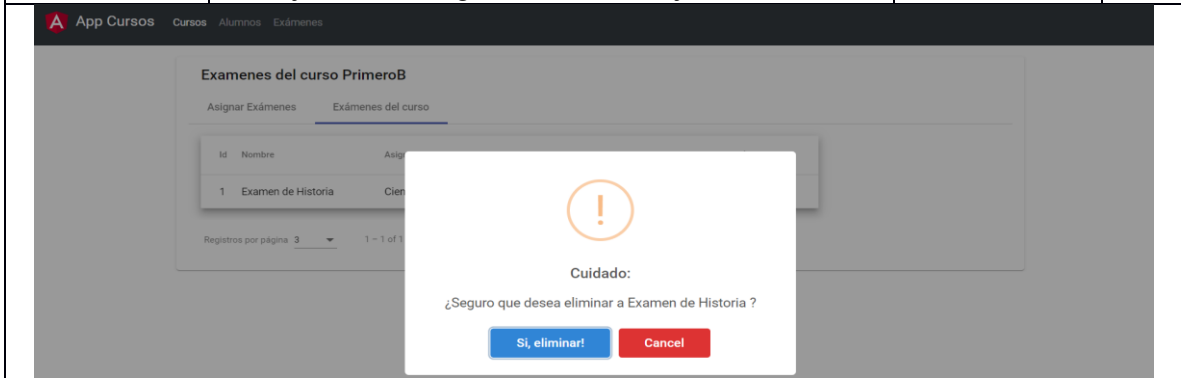
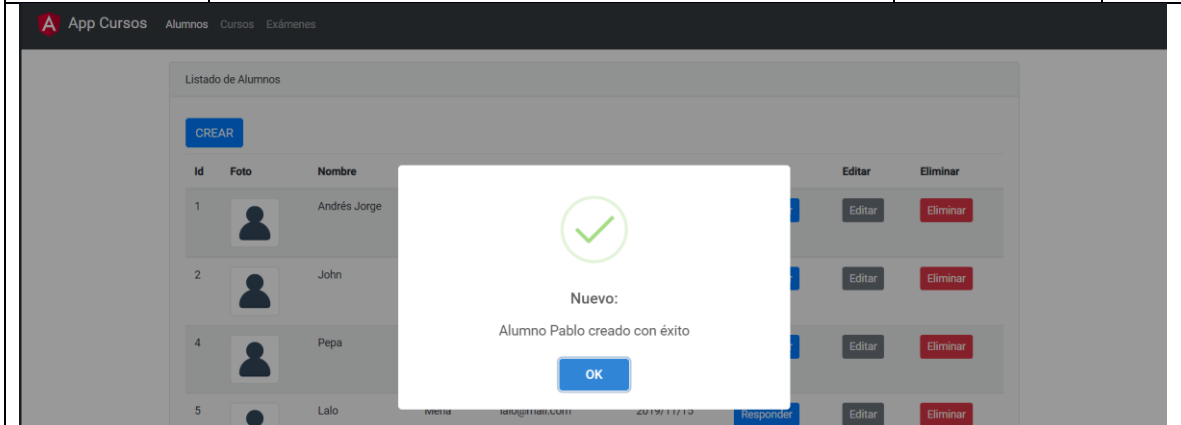


Tabla 34. Casos de Prueba N 001 - Registrar alumno

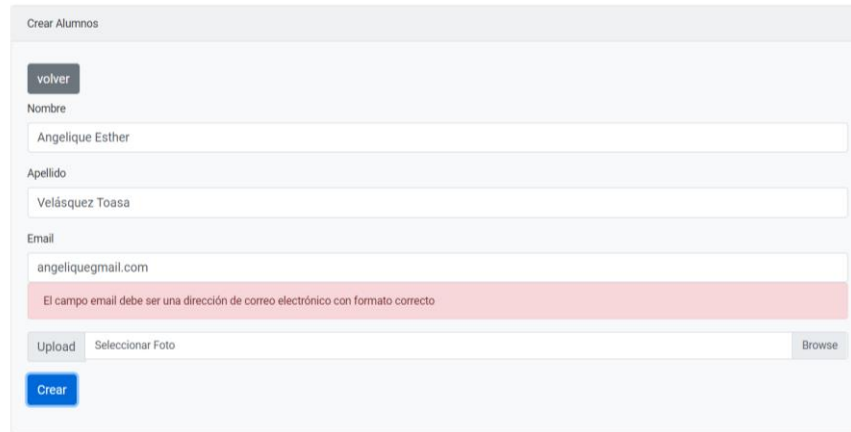
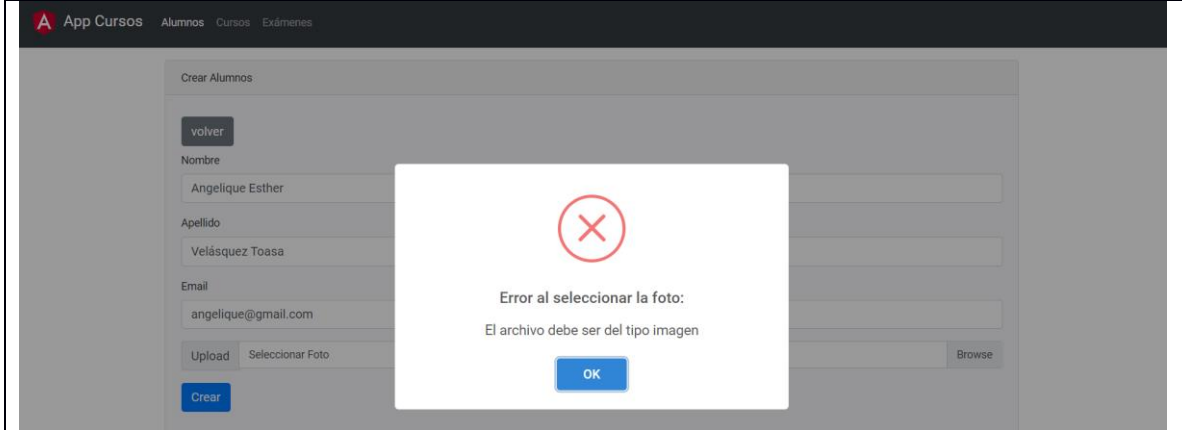
Nº:	001	Objetivo:	Registrar un Alumno	Usuario:	Docente
Descripción:	Como docente requiero registrar un alumno para ser asignado a un curso				
Condiciones de entrada:	<ul style="list-style-type: none"> • Estar autenticado • Abrir el formulario para agregar los datos del alumno 				
Entrada 1:	Nombre: Pablo, Apellido: Pérez, Email: pablito12@gmail.com foto:nsggasdhsag				
Resultado esperado:	Nuevo alumno Pablo creado con éxito	Evaluación de la prueba	Si		



Entrada 2:	Nombre: Pablo,		
-------------------	----------------	--	--

	Apellido: Pérez, Email: pablito1.utc.edu foto:		
--	------------------------------------------------------	--	--

Resultado esperado:	Mensaje de error “Formato incorrecto” por campos inválidos	Evaluación de la prueba	Si
----------------------------	------------------------------------------------------------	--------------------------------	----



Entrada 3:	Si existen campos vacíos		
-------------------	--------------------------	--	--

Resultado esperado:	Mensaje de error por campos requeridos	Evaluación de la prueba	Si
----------------------------	----------------------------------------	--------------------------------	----

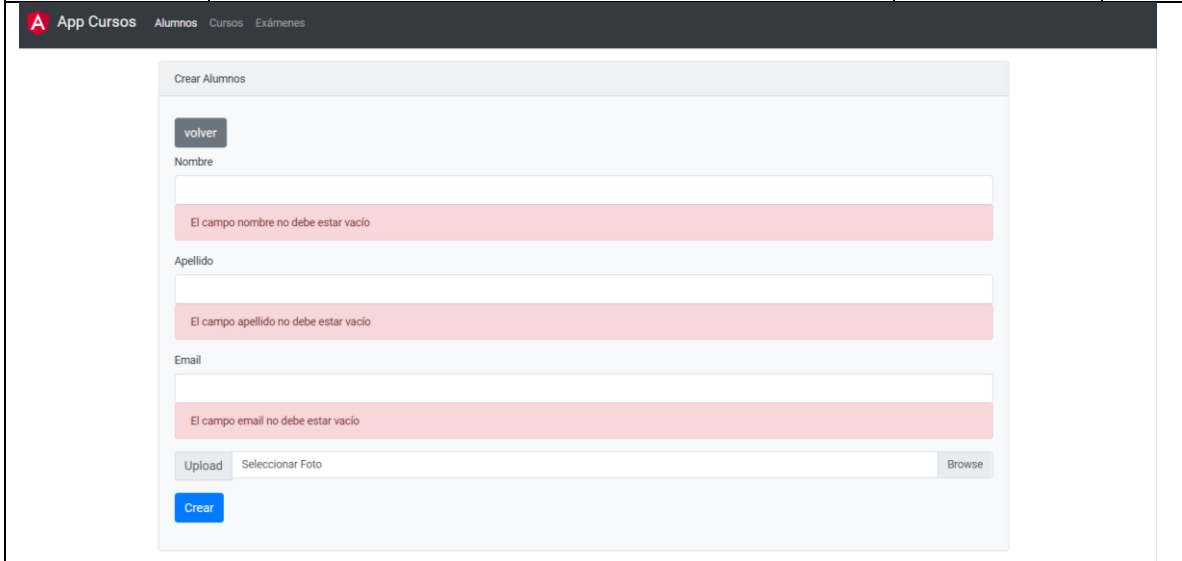


Tabla 35. Casos de Prueba N 002 - Mostrar alumno

N°:	002	Objetivo:	Mostrar datos del Alumno	Usuario:	Docente
Descripción:	Como docente requiero mostrar los datos del alumno para obtener una lista de alumnos existentes				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Abrir el formulario de Alumnos ● Estar autenticado ● Buscar los datos del alumno 				
Entrada 1:	No aplica				
Resultado esperado:	Muestra la lista de todos alumnos creados			Evaluación de la prueba	Si

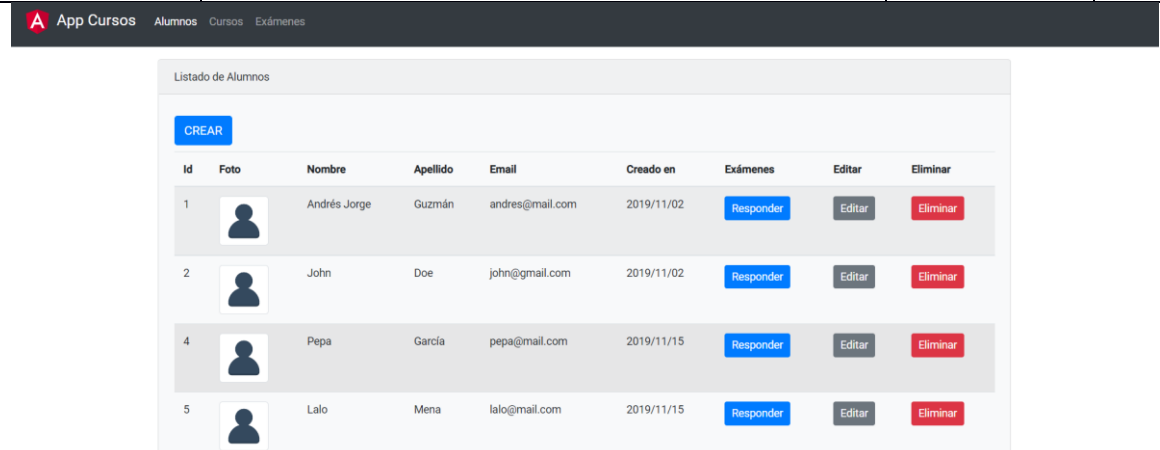
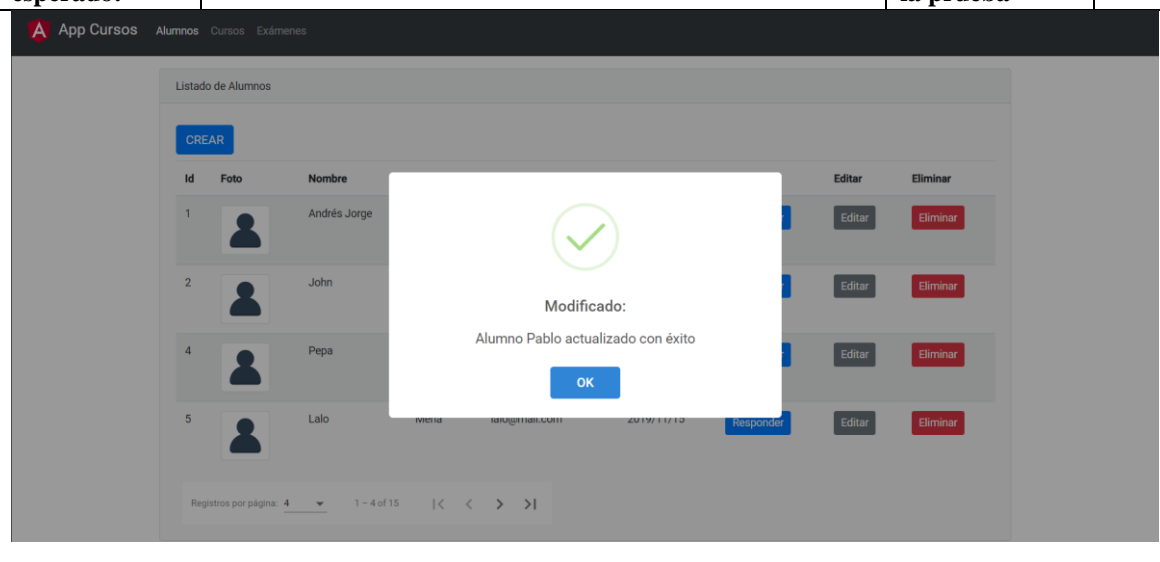


Tabla 36. Casos de Prueba N 003 - Modificar alumnos

N°:	003	Objetivo:	Editar información de un alumno	Usuario:	Docente
Descripción:	Como docente requiero actualizar los datos del alumno para tener la información más íntegra posible				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Estar autenticado ● Abrir el formulario para agregar los datos del alumno ● Dar clic en la opción editar del Alumno 				
Entrada 1:	Nombre: Pablo, Apellido: Pérez, Email: pablo12@utc.edu.ec				
Resultado esperado:	Nuevo alumno Pablo actualizado con éxito			Evaluación de la prueba	Si



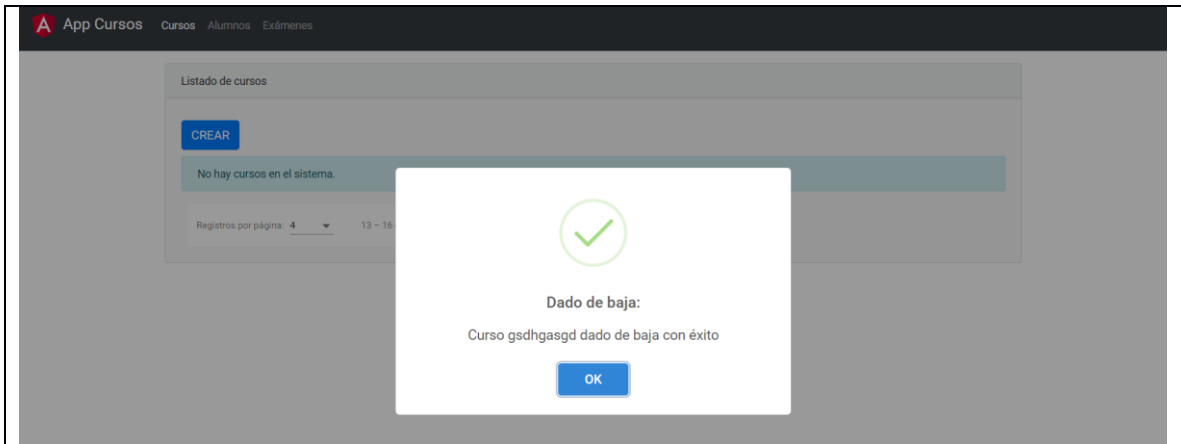
Entrada 2:	Nombre: Pablo, Apellido: Pérez, Email: pablito1.utc.edu		
Resultado esperado:	Mensaje de error “Formato incorrecto” por campos inválidos	Evaluación de la prueba	Si

Entrada 3:	Si existen campos vacíos		
Resultado esperado:	Mensaje de error por campos requeridos	Evaluación de la prueba	Si

15.3. Sprint 3

Tabla 37. Casos de Prueba N 04 Dar de baja un alumno

N°:	004	Objetivo:	Dar de baja un alumno	Usuario:	Docente
Descripción:	Como docente requiero dar de baja a un alumno para un mejor control de datos				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Estar autenticado ● Abrir el listado de alumnos ● Seleccionar el alumno 				
Entrada 1:	No aplica				
Resultado esperado:	Mensaje de alerta: “Alumno dado de baja exitosamente”	Evaluación de la prueba	Si		



Entrada 2:	No aplica		
Resultado esperado:	Si el docente decide no dar de baja a un alumno, presiona el botón “Cancelar”, vuelve a la pantalla principal de la sección “Cursos”. Mensaje de alerta “Seguro desea dar de baja”	Evaluación de la prueba	Si

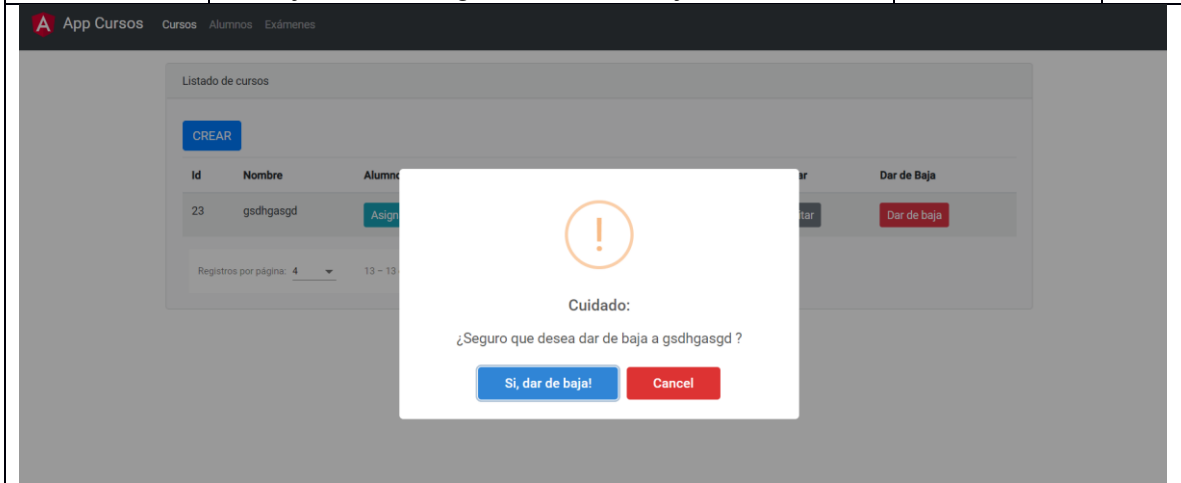
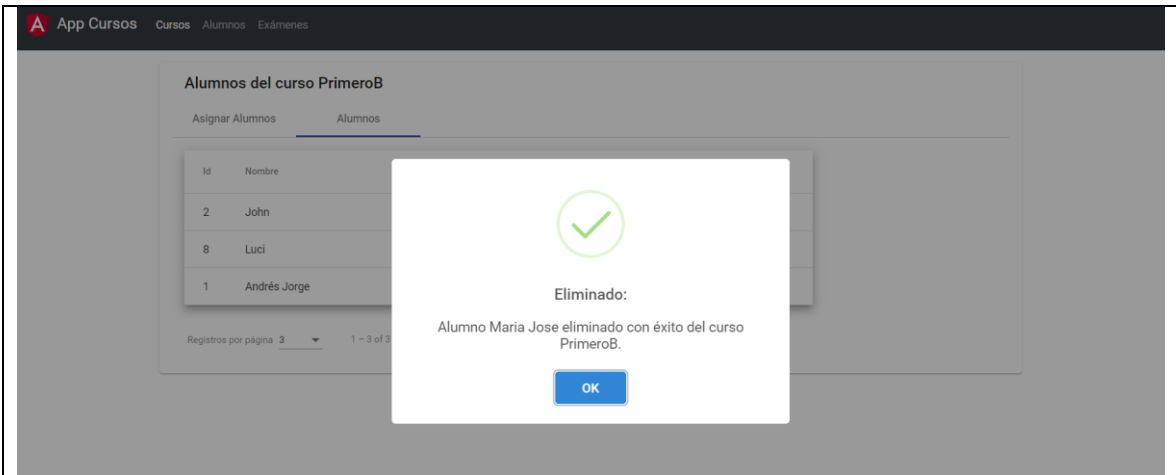


Tabla 38. Casos de Prueba N 005- Dar de baja alumno asignado

N°:	005	Objetivo:	Dar de baja un alumno asignado a un curso	Usuario:	Docente
Descripción:	Como docente requiero dar de baja a un alumno asignado a un curso para un mejor control de datos				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Abrir el formulario de cursos ● Estar autenticado ● Alumno asignado a un curso 				
Entrada 1:	No aplica				
Resultado esperado:	Mensaje de confirmación “Alumno María José dado de baja del curso Primero B ”	Evaluación de la prueba	Si		



Entrada 2:	No aplica		
Resultado esperado:	Mensaje de alerta “¿Seguro que desea eliminar a María José?”	Evaluación de la prueba	Si

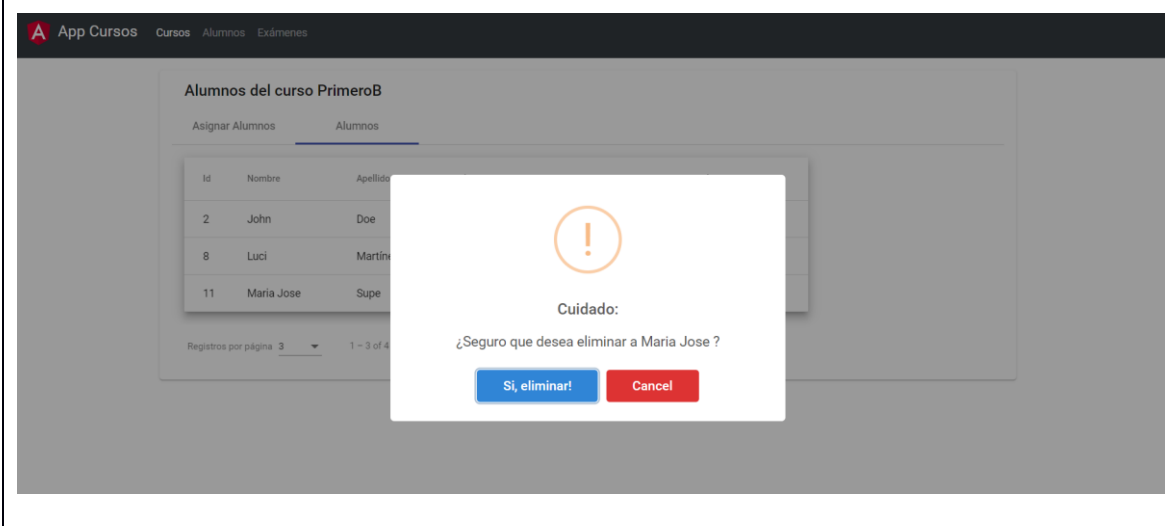
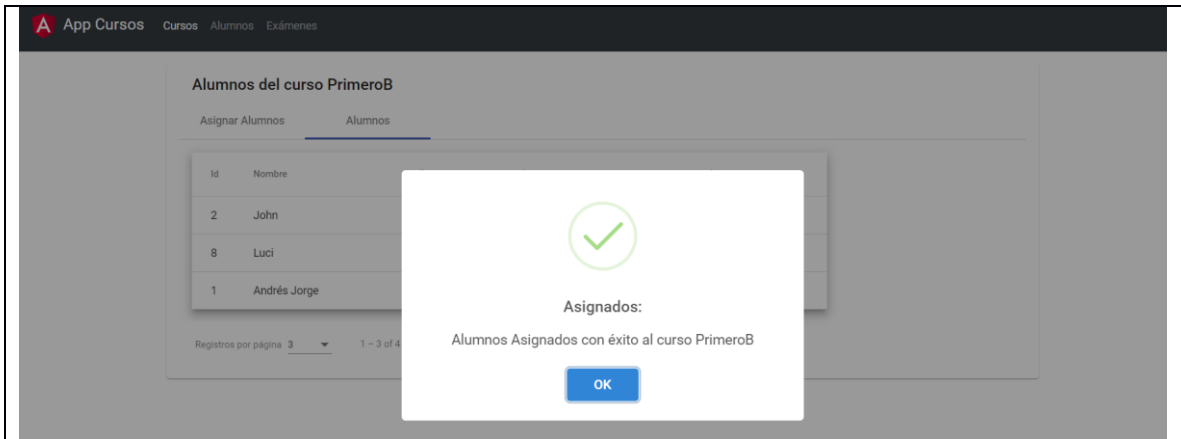


Tabla 39. Casos de Prueba N 010 - Asignar alumno

Nº:	010	Objetivo:	Asignar alumno	Usuario:	Docente
Descripción:	Como docente requiero asignar un alumno a un curso para que tenga acceso a sus exámenes				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Estar autenticado ● Listado de Cursos ● Escojo el curso ● Busco un alumno ● Asigno alumno 				
Entrada 1:	Curso: PRIMERO A Nombre: Pablo Apellido: Pérez				
Resultado esperado:	Alumno agregado con éxito al Curso Primero A	Evaluación de la prueba	Si		



Entrada 2:	Curso: PRIMERO B Nombre: Pablo Apellido: Pérez		
Resultado esperado:	Mensaje de error por información duplicada “No se puede asignar al alumno ya está asociado a otro curso”	Evaluación de la prueba	Si

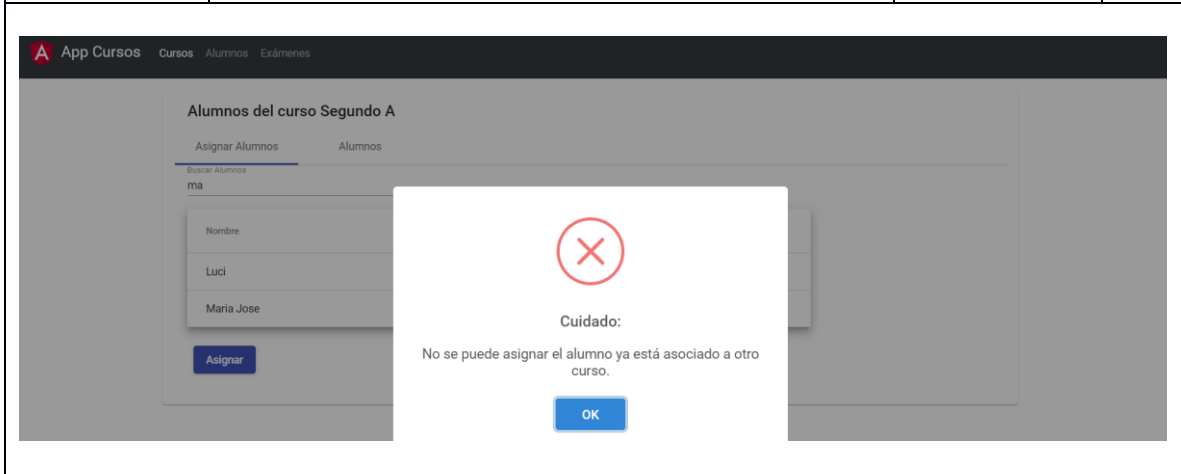
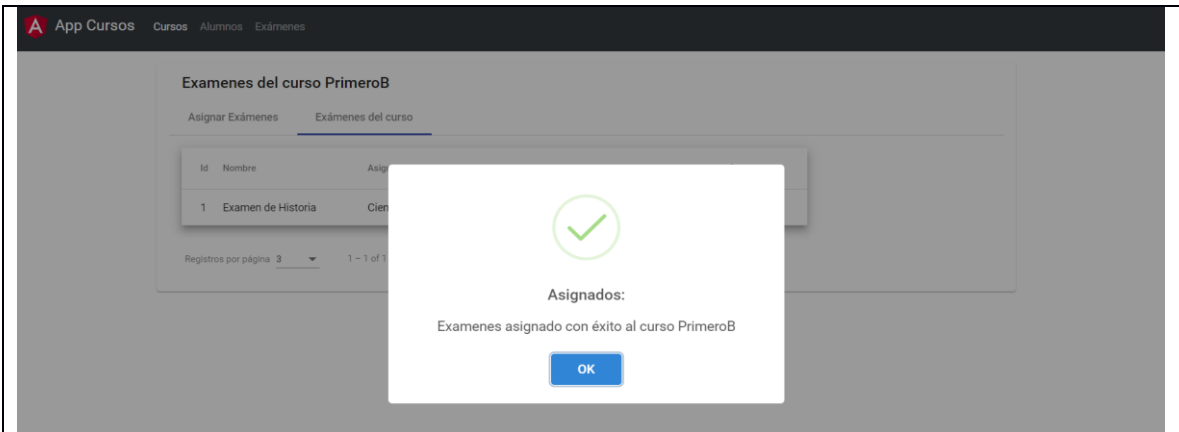


Tabla 40. Casos de Prueba N 011 - Asignar examen

Nº:	011	Objetivo:	Asignar examen	Usuario:	Docente
Descripción:	Como docente requiero asignar un examen a un curso para que los alumnos puedan responder				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Estar autenticado ● Listado de cursos ● Escojo el curso ● Busco un examen ● Asignación de examen 				
Entrada 1:	Curso: PRIMERO A Nombre: Examen de Matemáticas Apellido: Aritmética				
Resultado esperado:	Examen asignado con éxito a Primero A			Evaluación de la prueba	Si



Entrada 2:	Curso: PRIMERO A Nombre: Examen de Matemáticas Apellido: Aritmética		
Resultado esperado:	Mensaje de error por información duplicada “El examen ya está asignado al curso”	Evaluación de la prueba	Si

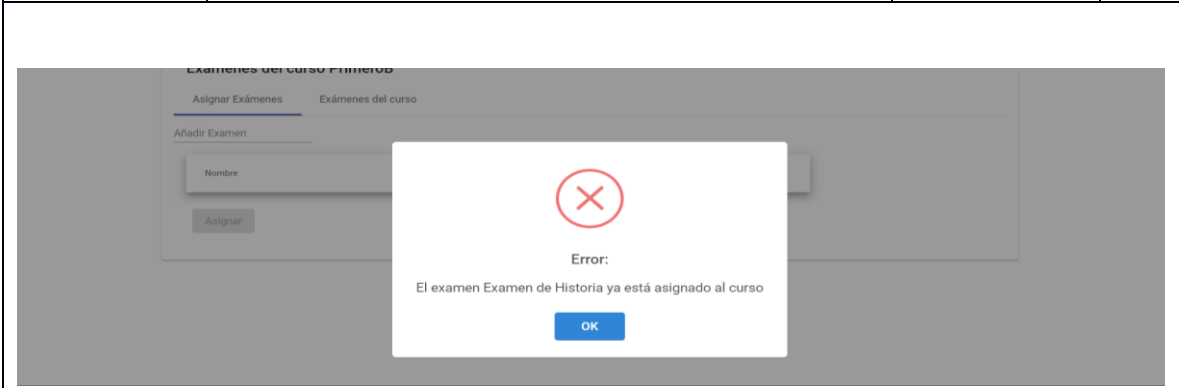


Tabla 41. Casos de Prueba N 017- Ver exámenes respondidos

N°:	017	Objetivo:	Ver respuestas del examen	Usuario:	Docente
Descripción:	Como alumno requiero ver los exámenes respondidos para observar las respuestas				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Abrir el formulario de modificación de usuarios ● Estar autenticado ● Examen respondido por el alumno 				
Entrada 1:	No aplica				
Resultado esperado:	Muestra los exámenes respondidos con sus preguntas	Evaluación de la prueba	Si		

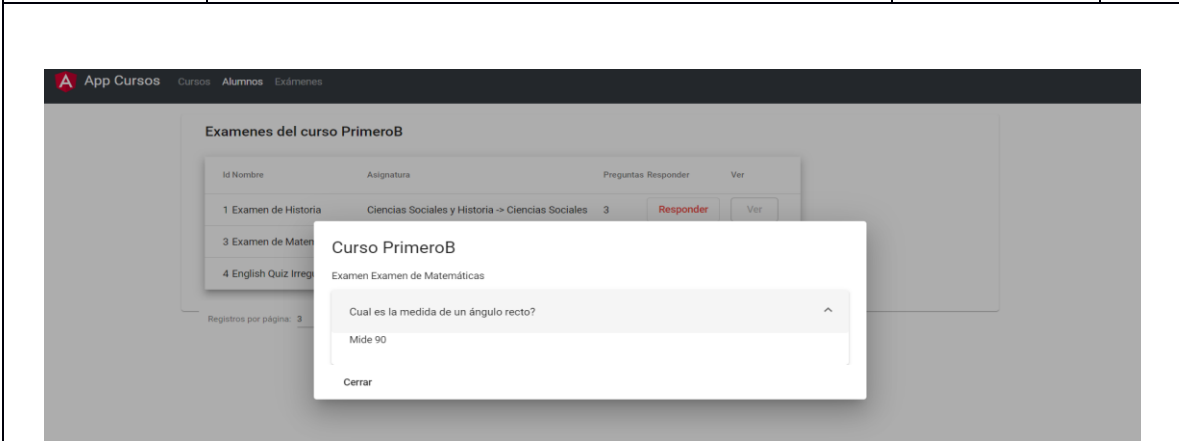
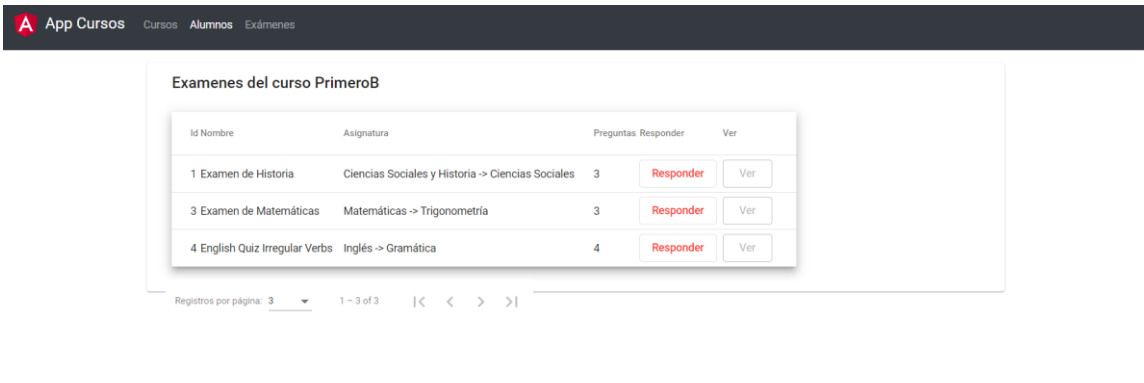


Tabla 42. Casos de Prueba N 019 - Ver exámenes asignados

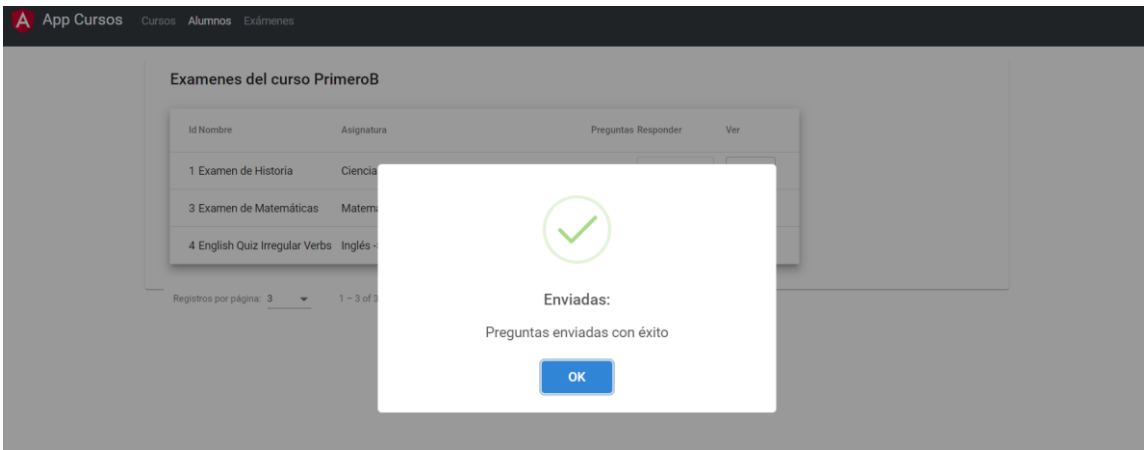
Nº:	019	Objetivo:	Ver exámenes asignados	Usuario:	Alumno
Descripción:	Como alumno deseo ver los exámenes asignados para poder responder				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Abrir el formulario de Exámenes ● Estar autenticado ● Tener asignado exámenes ● Seleccionar los exámenes asignados 				
Entrada 1:	No aplica				
Resultado esperado:	Muestra la lista de todos los exámenes asignados	Evaluación de la prueba	Si		



The screenshot shows the 'App Cursos' interface. At the top, there is a navigation bar with 'App Cursos', 'Cursos', 'Alumnos', and 'Exámenes'. Below this, the title 'Exámenes del curso Primerob' is displayed. A table lists the exams with columns for 'Id Nombre', 'Asignatura', 'Preguntas', 'Responder', and 'Ver'. The table contains three rows: '1 Examen de Historia' (Ciencias Sociales y Historia -> Ciencias Sociales, 3 questions), '3 Examen de Matemáticas' (Matemáticas -> Trigonometría, 3 questions), and '4 English Quiz Irregular Verbs' (Inglés -> Gramática, 4 questions). Each row has a red 'Responder' button and a grey 'Ver' button. At the bottom of the table, there is a pagination control showing 'Registros por página: 3' and '1 - 3 of 3'.

Tabla 43. Casos de Prueba N 018

Nº:	018	Objetivo:	Responder preguntas	Usuario:	Alumno
Descripción:	Responder las preguntas del examen en el curso asignado				
Condiciones de entrada:	<ul style="list-style-type: none"> ● Estar autenticado ● Listado de exámenes ● Escoger examen ● Responder preguntas 				
Entrada 1:	Respuesta: Mide 90 grados				
Resultado esperado:	Preguntas enviadas con éxito	Evaluación de la prueba	Si		



The screenshot shows the 'App Cursos' interface with a success message overlay. The background is dimmed, showing the same exam list as in the previous screenshot. A white dialog box with a green checkmark icon is centered on the screen. The text in the dialog box reads: 'Enviadas: Preguntas enviadas con éxito' and 'OK'.

ANEXO 3: TABLA DE RESULTADOS

En la siguiente tabla se describe detalladamente los resultados obtenidos del cálculo de acoplamiento y grado de falta de cohesión.

Tabla 44. Detalle de métricas e interpretación de resultados

Expresión	Interpretación del resultado	
$AIS_1=1$ $AIS_2=0$ $AIS_3=1$ $AIS_4=1$ $AisT = 1.73$	<p>AIS representa el número de otros microservicios que tienen una dependencia a este microservicio.</p> <p>Según [21] menciona que la importancia absoluta permite obtener el número de peticiones que tiene cada microservicio. Tomando como referencia este valor se puede decir si el microservicio no tiene mayor relación con otros para ejecutar sus tareas.</p> <p>Con lo antes mencionado se tiene que los microservicios uno, tres y cuatro solo tienen una petición y el microservicio 2 no tiene ninguna petición de otros microservicios para cumplir sus tareas, el valor de independencia total del sistema muestra que los microservicios no están con mayor número de relaciones y cada microservicio se desarrollará de forma independiente.</p>	
$ADS_1=2$ $ADS_2=2$ $ADS_3=0$ $ADS_4=0$ $AdsT = 2.83$	<p>ADS representa el número de otros microservicios de los que depende este microservicio, es decir, de los que invoca al menos una operación.</p> <p>Según [21] menciona que la dependencia absoluta permite obtener el número de peticiones que realiza cada microservicio a otro para cumplir una tarea específica microservicio tomando como referencia este valor se puede decir el número de funcionalidades que dependen de otras para ejecutar su tarea.</p> <p>En base a esto se puede decir que el microservicio tres y cuatro no tiene ninguna funcionalidad que necesite de otra para cumplir con su tarea a comparación de microservicio uno y dos existe solo dos funcionalidades en cada uno que necesitan de otro microservicio, el valor total de dependencia absoluta total del sistema muestra que los microservicios no están con mayor número de relaciones y cada microservicio se desarrollará de forma independiente.</p>	
$SIY_1=0/4$ $SIY_2=0/4$ $SIY_3=0/4$ $SIY_4=0/4$ $SiyT = 0$	<p>SIY representa el número de microservicios de los que dependen bidireccionalmente unos de otros.</p> <p>Según [21] menciona que el valor de interdependencia es un mal indicador por lo que debe ser reducido a cero, en base a esto se puede decir que se obtuvo un valor igual a 0 de interdependencia lo que reduce la complejidad al desarrollar el prototipo y realizar cambios sin que afecte su funcionamiento.</p>	
$LC_1=1/4$ $LC_2=0/4$ $LC_3=1/4$ $LC_4=2/4$	<p>Según [21] menciona que la falta de cohesión mide la dependencia de forma bidireccional esto quiere decir que un microservicio dependa de las funcionalidades de varios microservicios para ejecutar una tarea específica. Como se puede observar cada microservicio.</p>	
$CpT = 3.32$	<p>Alto</p>	<p>Bajo</p> <p>X</p> <p>En [35] menciona que, si mayor es el número de dependencias entre microservicios, mayor es el nivel de acoplamiento, un microservicio que tenga alta cohesión debe funcionar por sí solo sin depender en un alto grado de otros microservicios y no tener interdependencias con los otros microservicios.</p>
$CohT = 0.60$	<p>X</p>	<p>En base a los valores mencionados se obtuvo un bajo acoplamiento ya que se realizó un agrupamiento de historias de usuario según el funcionamiento de microservicios disminuyendo la dependencia en cada microservicio y aumentando la cohesión, se obtuvo un valor bajo de grado de falta de cohesión ya que los microservicios no están fuertemente conectados entre sí y pueden desarrollarse de forma independiente, cada microservicio tiene sus funcionalidades.</p>

En Tabla 46, se describe detalladamente los resultados obtenidos del cálculo de la complejidad del caso de estudio.

Tabla 45. Detalle de las métricas e interpretación de resultados

Expresión	Interpretación del resultado			
$ms_1 = 2$ $ms_2 = 2$ $ms_3 = 0$ $ms_4 = 0$ $Avg. Calls = 4$	Según [21] menciona que las llamadas son las invocaciones que realiza un microservicio a otro microservicio, donde el microservicio uno y dos tienen dos llamadas cada uno, el microservicio tres y cuatro no tiene a ninguna llamada de ningún otro microservicio.			
$ms_1 = 1$ $ms_2 = 0$ $ms_3 = 1$ $ms_4 = 2$ $Request = 4$	Según [21] menciona que las solicitudes son las invocaciones que realiza un microservicio a otro microservicio, donde el microservicio uno tiene una solicitud, el microservicio dos ni tiene ninguna solicitud, el microservicio tres tiene una solicitud, finalmente el microservicio cuatro no tiene dos solicitudes.			
$p_i = 16$	Este valor corresponde a la suma de los puntos de historias utilizando la sucesión de Fibonacci.			
$Max = 8$	Una vez agrupado los puntos de historia se obtienen el valor mayor del punto de historia asignado a las historias de usuario para este caso tenemos el 8 como mayor valor de punto de historia.			
$WsicT = Max(7)$	Este valor corresponde al mayor valor de historias de usuario que va a tener un microservicio, es decir el tamaño máximo del microservicio.			
$pf = 0$	Este valor corresponde al número de llamadas usadas secuencialmente hacia otros microservicios, es decir que un microservicio realiza múltiples llamadas a otros microservicios para cumplir una tarea específica.			
$SIY = 0$	Según [21] menciona que el valor de interdependencia es un mal indicador por lo que debe ser reducido a cero, se puede decir que para este caso de estudio se obtuvo un valor bajo de interdependencia lo que reduce la complejidad al desarrollar el prototipo y realizar cambios sin que afecte su funcionamiento.			
$Cx_0 = 2.$	Este valor corresponde al caso base en el que la aplicación tiene un microservicio, una historia de usuario con un punto de historia estimado, este es el caso más simple de desarrollar.			
	Alto	Medio	Bajo	Interpretación del resultado
$CxT = 82.30$			X	Para este caso 82.30 corresponde a que la complejidad total del prototipo va ser baja teniendo como referencia el caso base según [21], esto quiere decir que la dificultad de entender, desarrollar y mantener va hacer menos complejo.

ANEXO 4: DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

Tabla 46. Definiciones, acrónimos y abreviaturas

ABREVIATURA	SIGNIFICADO
MSBA	Arquitectura basada en microservicios
AIS	Importancia absoluta del microservicio
ADS	Dependencia absoluta del microservicio
SIY	Interdependencia del microservicio
COH	Grado de falta de cohesión
LC	Falta de cohesión
WSICT	Granularidad

ANEXO 5: CARTA A EXPERTOS PARA EVALUACIÓN DE LA ENCUESTA

Latacunga, Agosto 2022

Ing.

Santiago Xavier Merlo Vega

Presente.-

Tenemos el agrado de dirigirnos a Ud., para saludarlo(a) cordialmente y a la vez solicitarle su valiosa colaboración como Juez Experto para revisar el contenido del instrumento que pretendemos utilizar, con la finalidad de determinar la validez del modelo de optimización aplicado en el diseño del prototipo de la aplicación. Considerando la siguiente escala de medición marcar con una X según el grado de evaluación de los indicadores.

Agradecemos anticipadamente su colaboración y estamos seguras que su opinión y criterio de experto servirán para los fines propuestos.

Atentamente:



Salazar Ninasunta Alexandra Jeaneth
C.I.: 0550007595



Toasa Supe María José
C.I.: 1804664819

Identificación del experto

Tabla 47. Identificación del experto

Nombre y apellidos	Santiago Xavier Merlo Vega
N° de Identificación	0502757115
Grado Académico	Ingeniero en sistemas, tercer nivel
Lugar de Trabajo	Grupo BABEL
Cargo que desempeña	Gerente regional de innovación tecnológica
Experiencia	13 Años
e-mail	santiago.merlo@grupobabel.com
Teléfono o celular	0999689909
Fecha de la validación (día, mes y año):	17/08/2022
Firma	<hr/>

Encuesta

Instrumento para la validar la propuesta

Objetivo:

El siguiente instrumento tiene como objetivo validar la propuesta del modelo de optimización mediante la aplicación de la encuesta y así conocer la ponderación del grupo de juicio de expertos.

Instrucciones:

Por favor lea detenidamente cada pregunta y responda con la mayor sinceridad y confianza, Si no entiende alguna pregunta o alguna palabra, pídale a la persona que está a cargo de la encuesta que le explique.

En una escala del 1 al 5. ¿Cuál cree que es la factibilidad del siguiente modelo de optimización?, siendo 1 totalmente desacuerdo, 2 en desacuerdo, 3 indiferente ,4 de acuerdo, 5 totalmente de acuerdo considerado el valor 1 más deficiente y 5 el valor más eficiente.

Nº	Pregunta	Ponderación				
		1	2	3	4	5
1	¿Considera usted que el modelo de optimización propuesto es eficiente para el diseño de aplicaciones basadas en microservicios?					
2	¿Considera usted que el modelo de optimización propuesto en este trabajo de investigación ayudaría en el diseño de otras aplicaciones basadas en microservicios?					
3	¿Considera usted que el modelo arquitectónico está de acuerdo al modelo de optimización propuesto?					
4	¿Considera usted que el prototipo de la aplicación está de acuerdo al modelo arquitectónico ?					
5	¿Considera usted que las métricas de importancia absoluta, dependencia absoluta e interdependencia de los microservicios permitieron reducir el acoplamiento?					
6	¿Considera usted que la métrica de grado de falta de cohesión permitió reducir la dependencia entre los microservicios?					
7	¿Considera usted que las métricas de rendimiento, los puntos de historia y la complejidad cognitiva permitió reducir la complejidad?					

ANEXO 6: HOJA DE VIDA DEL EXPERTO

HOJA DE VIDA

DATOS PERSONALES

Nombre completo: **Santiago Xavier Merlo Vega**
Nacionalidad: ecuatoriana
Documento de identificación: 0502757115
Profesión: Ingeniero en Sistemas
Fecha de nacimiento: 19 de mayo 1987

EXPEDIENTE ACADÉMICO/CERTIFICACIONES/CURSOS

- **Ingeniero en Sistemas Informáticos y Computacionales**
Universidad Técnica de Cotopaxi.
Ecuador
2015
- **Scrum Fundamentals Certified**
SCRUMstudy
2016
- **Scrum Máster**
SCRUMstudy
2019
- **Six Sigma Yellow Belt**
SCRUMstudy
2016
- **“Microsoft Certified Trainer” MCT**
Microsoft
- **Microsoft Certified: Azure Developer Associate**
Microsoft
- **Microsoft Certified: Azure DevOps Engineer Expert**
Microsoft
- **Certified SAFe 5 Agilist**
Scaled Agile
- **Oracle Cloud Infrastructure Architect Associate**
Oracle
- **Oracle Cloud Infrastructure Architect Professional**
Oracle
- **Oracle Cloud Infrastructure 2021 Cloud Operations Associate**
Oracle

- **“Partner Practice Enablement Bootcamp” SQL 2014**
Microsoft Costa Rica
2014.

EXPERIENCIA LABORAL

Cargo: Director de Innovación tecnológica

Periodo: Noviembre 2021 – actualidad

Nombre de empresa: Grupo Babel Regional.

Cargo: Chief Technology Officer (CTO)

Periodo: Febrero 2019 – Octubre 2021

Nombre de empresa: Grupo Babel Regional.

Cargo: Gerente Técnico

Periodo: Febrero 2017 – Enero 2019

Nombre de empresa: Soluciones informáticas Babel del Ecuador.

Cargo: Especialista DevOps

Periodo: Noviembre 2014 – Actualidad

Nombre de empresa: Soluciones informáticas Babel del Ecuador.

Cargo: Consultor de Software

Periodo: Junio 2013 – Octubre 2014

Nombre de empresa: Soluciones informáticas Babel del Ecuador.

Cargo: Pasante

Periodo: Marzo 2013 – Mayo 2013

Nombre de empresa: Soluciones informáticas Babel del Ecuador.

Cargo: Componente Informático

Periodo: Agosto 2009 – Marzo 2011

Nombre de empresa: Vicepresidencia de la república del Ecuador



PERFIL

Desarrollador de aplicaciones para automatización procesos institucionales de gobiernos seccionales, facturación electrónica. Consultor de sistemas de gestión documental.

Director ejecutivo de desarrollo en fingads.net
Director ejecutivo de desarrollo en daes.cat
Ex catedrático universitario

Apasionado por entender los negocios de los clientes y como poder ayudarles a crecer.

CONTACTO

TELÉFONO:
+593997595844

LINKEDIN:
<https://www.linkedin.com/in/diego-mu%C3%B1oz-a4795931/>

CORREO ELECTRÓNICO:
diegoamandomi@gmail.com

ACTIVIDADES E INTERESES

Ciclismo
Buceo
Natación
Pesca deportiva

IDIOMAS

Español - Nativo
Inglés - Buen conocimiento práctico

DIEGO MUÑOZ

Desarrollador de aplicaciones para automatización procesos institucionales de gobiernos seccionales, facturación electrónica y Consultor de sistemas de gestión documental.

EXPERIENCIA LABORAL

Director ejecutivo de desarrollo en daes.cat

2017 – Actualidad

Daes es una compendio de distintas herramientas informáticas y de sistemas, principalmente de gestión de clientes, tributación y servicios a corporaciones del área financiera.

Director ejecutivo de desarrollo en fingads.net

2012 – Actualidad

Creador del sistema financiero contable para las juntas parroquiales rurales fingads.net en sociedad como coautor, el sistema permite generar asientos contables y reportes financieros a través de la gestión de procesos administrativos, permitiendo a las juntas manejar correctamente la contabilidad.

Especialista 4

Gobierno autónomo Descentralizado de la Provincia de Sucumbíos

2008 – Actualidad

Administrador y desarrollador del área de sistemas

Profesor universitario

Escuela Superior Politécnica Ecológica Amazónica

2010 - 2012

Profesor a tiempo completo de desarrollo de aplicaciones y gestión de base de datos

Técnico de operaciones

Gobierno Provincial de Sucumbíos

2005 - 2008

Asistencia técnica a procesos administrativos, mantenimiento de computadores y soporte de software

FORMACIÓN ACADÉMICA

Escuela Superior Politécnica de Chimborazo

Abril 2007 – Ingeniero en sistemas Informáticos

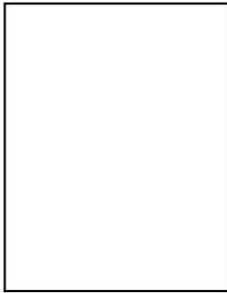
Escuela Superior Politécnica de Chimborazo

Julio 2022 – Magister en Archivística y sistemas de gestión documental

CARACTERÍSTICAS Y COMPETENCIAS CLAVE

- Entendimiento del problema a resolver
- Capacidad para proponer soluciones alternativas
- Trabajo bajo presión
- Transferencia de soluciones y de la tecnología para hacerlo
- Minimizar dependencia posterior

JUAN MAURICIO SALGADO QUIMBITA



Datos Personales

Dirección: AV. GRL. MIGUEL ITURRALDE S/N y N/A
Teléfono (s): 032262312 - 0969685038 - 032990018 - 0987480904
Cédula de identidad: 0502410525
Correo electrónico: juan_salgado1986@hotmail.com
Ciudad / Provincia / País: LATACUNGA / COTOPAXI / ECUADOR

Instrucción Formal

INGENIERO EN INFORMATICA Y SISTEMAS COMPUTACIONALES	TERCER NIVEL
(5 AÑOS)	UNIVERSIDAD TECNICA DE COTOPAXI
MAGISTER EN INGENIERIA DE SOFTWARE	EGRESADO
(2 AÑOS)	UNIVERSIDAD DE LAS FUERZAS ARMADAS (ESPE)

Experiencia:

2019/12/05

ANALISTA DE TECNOLOGÍAS DE LA INFORMACIÓN / EMPRESA PÚBLICA DE ASEO Y

- * IDENTIFICAR LAS NECESIDADES INSTITUCIONALES Y DAR SOLUCIÓN A LOS REQUERIMIENTOS INHERENTES A LA GESTIÓN TECNOLÓGICA
- * COORDINAR CON LOS PROCESOS INSTITUCIONALES, LA IMPLEMENTACIÓN DE LOS ENLACES INFORMÁTICOS ENTRE LOS SISTEMAS QUE POSEE LA EPAGAL
- * EJECUTAR ACCIONES PARA EL CUMPLIMIENTO DE LAS METAS ESTRATÉGICAS, OPERATIVAS E INDICADORES, A EFECTOS DE DETERMINAR LA PRODUCTIVIDAD DE LOS PROCESOS Y LA GENERACIÓN PERIÓDICA DE REPORTE INSTITUCIONALES
- * BRINDAR SOPORTE TÉCNICO, OPERATIVO Y PERSONALIZADO PARA LA SOLUCIÓN DE PROBLEMAS PRESENTADOS A LOS USUARIOS EN HARDWARE, SOFTWARE Y APLICACIONES ESTÁNDARES DE LA EPAGAL
- * FORMULAR Y EJECUTAR LOS PLANES OPERATIVOS (PLAN INFORMÁTICO, PLAN DE MANTENIMIENTO PREVENTIVO Y CORRECTIVO, PLAN DE CONTINGENCIAS TECNOLÓGICAS) Y CONTROLAR SU CUMPLIMIENTO
- * PROPONER MEJORAS DE LOS PROCESOS INSTITUCIONALES CON BASE EN EL DESARROLLO DE APLICACIONES INFORMÁTICAS, LA UTILIZACIÓN DE NUEVOS SISTEMAS INFORMÁTICOS Y HERRAMIENTAS TECNOLÓGICAS INCORPORANDO ESTÁNDARES INTERNACIONALES
- * GENERAR PROCEDIMIENTOS DE INNOVACIÓN TECNOLÓGICA, EN FUNCIÓN A LOS SISTEMAS DE GESTIÓN ORGANIZACIONAL (EMPRESARIAL, AMBIENTAL Y DE INFORMACIÓN)
- * LAS DEMÁS RESPONSABILIDADES INHERENTES AL CARGO, QUE LE FUEREN ENCOMENDADAS POR SUS SUPERIORES, Y AQUELLAS QUE SEAN DESIGNADAS POR LAS NORMALES LEGALES CORRESPONDIENTES
- * GARANTIZAR LA DISPONIBILIDAD PERMANENTE DE LOS SERVICIOS DE SOPORTE Y MANTENIMIENTO, A TRAVÉS DE LA ADECUADA ADMINISTRACIÓN DE LOS SISTEMAS INFORMÁTICOS, APLICATIVOS Y REDES LOCALES, REMOTAS DE VOZ Y DATOS

2010/05/24 2013/01/08

INGENIERO DEL AREA DE SISTEMAS / FARMACIAS CRUZ AZUL

* ENCARGADO ADMINISTRADOR DE LA RED DE LA CADENA DE FARMACIAS CRUZ AZUL DE LATACUNGA, MANTENIMIENTO PREVENTIVO Y CORRECTIVO DE COMPUTADORAS, ETC.

2010/05/24 2019/12/04

ING. SISTEMAS / CAROLINA RIOS CIA. LTDA.

* ENCARGADO DEL ÁREA DE SISTEMAS, MANTENIMIENTO PREVENTIVO Y CORRECTIVO DE COMPUTADORAS, CÁMARAS DE SEGURIDAD, BODEGA, COMPRAS PUBLICAS, INVENTARIOS, FACTURACIÓN, MANEJO DE BASE DE DATOS, ETC.

2009/09/14 2010/04/07

ADMINISTRADOR / ANDESOFIT - LATACUNGA

* MANTENIMIENTO DE LA RED Y COMPUTADORAS, DIGITADOR, ADMINISTRADOR DEL CENTRO DE COMPUTO.

Idioma:

INGLÉS	Nivel Hablado:	BÁSICO
	Nivel Escrito:	INTERMEDIO
ESPAÑOL	Nivel Hablado:	NATIVO
	Nivel Escrito:	NATIVO

Capacitación:

TALLER (40 horas)	CABLEADO ESTRUCTURADO Y REDES DE DATOS BLUE IT EDUCACIÓN BÁSICA/CURSOS
SEMINARIO (20 horas)	ADMINISTRACIÓN FIREWALL SOPHOS XG CNT EDUCACIÓN BÁSICA/CURSOS
SEMINARIO (24 horas)	LIDERAZGO Y TRABAJO EN EQUIPO EN EL CONTROL CONTRALORÍA GENERAL DEL ESTADO EDUCACIÓN BÁSICA/CURSOS
TALLER (36 horas)	SOFTWARE LIBRE. (INSTALACIÓN LINUX DEBIAN) ESPE INFORMÁTICA SOFTWARE

PASANTIA (132 horas)	CONTROL Y ADMINISTRACIÓN DE LOS USUARIOS DE LA JUNTA JUNTA ADMINISTRADORA DE AGUA POTABLE INFORMÁTICA SOFTWARE
SEMINARIO (9 horas)	OPERADOR DEL SISTEMA DE CONTRATACIÓN PÚBLICA - SERVICIO NACIONAL DE CONTRATACIÓN PÚBLICA EDUCACIÓN BÁSICA/CURSOS
SEMINARIO (7 horas)	APRENDE JOOMLA 3.9 UDEMY EDUCACIÓN BÁSICA/CURSOS
TALLER (10 horas)	WORKSHOP DE VINCULACIÓN DEL PROGRAMA DE MAESTRÍA EN UNIVERSIDAD DE LAS FUERZAS ARMADAS CIENTÍFICO/INVESTIGACIÓN
SEMINARIO (40 horas)	USER EXPERIENCE & USER INTERFACE DESING UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE INFORMÁTICA SOFTWARE
SEMINARIO (30 horas)	PAQUETES UTILITARIOS UNIVERSIDAD TÉCNICA DE COTOPAXI INFORMÁTICA SOFTWARE
SEMINARIO (20 horas)	PROGRAMACION PHP UNIVERSIDAD TÉCNICA DE COTOPAXI INFORMÁTICA SOFTWARE
SEMINARIO (36 horas)	MANTENIMIENTO PREVENTIVO Y CORRECTIVO DE UNIVERSIDAD TÉCNICA DE COTOPAXI INFORMÁTICA SOFTWARE
JORNADA (20 horas)	CERTIFICACION B2 IDIOMA INGLÉS UNIVERSIDAD TÉCNICA DE MANABÍ EDUCACIÓN/UNIVERSIDAD

Acciones Afirmativas

Nota: Las acciones afirmativas se sumarán, únicamente, a aquellos postulantes que cumplan con los requisitos para recibirlos y siempre que hubieran obtenido la calificación mínima del setenta por ciento (70%) en el puntaje de

Autodeterminación étnica: MESTIZO/A

Discapacidad: NO

Enfermedad catastrófica: NO

A cargo de familiar con NO

A cargo de familiar con NO

Migrante Ex-Servidor Público: NO

Residente provincia de NO

Referencias Personales

ING. JOSE CAIZA

0984552122

jrcaiza@espe.edu.ec

ING. MARCELO JESUS QUIMBITA QUIMBITA

0979313448

remasterload@hotmail.com

ING FRANKLIN JAVIER MONTALUISA YUGLA

0988518645

fjmontaluisa@espe.edu.ec