



Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Portada

UNIVERSIDAD NACIONAL DE LOJA

FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS
Y LOS RECURSOS NATURALES NO RENOVABLES

CARRERA DE INGENIERÍA EN SISTEMAS

Estudio comparativo de metodologías de desarrollo de software para
aplicaciones móviles

Trabajo de titulación previo a
la obtención del título de
ingeniero en sistemas.

Autor:

Andy Aron Camacho Herrera

Director:

Ing. Edison Leonardo Coronel Romero, Mg.Sc.

LOJA – ECUADOR
2022

Certificación del Trabajo de Titulación

Ing. Edison Leonardo Coronel Romero, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN.

CERTIFICO:

Que el egresado **Andy Aron Camacho Herrera** autor del presente trabajo de titulación, cuyo tema versa sobre **“Estudio Comparativo de Metodologías de Desarrollo de Software para aplicaciones móviles”**, ha sido dirigido, orientado, discutido bajo mi asesoramiento y ha sido culminado al 100%, reúne a satisfacción los requisitos exigidos en una investigación de este nivel por lo cual autorizo su presentación y sustentación.

Loja, 17 de septiembre del 2021

Ing. Edison Leonardo Coronel Romero, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Yo **Andy Aron Camacho Herrera**, declaro ser autor del presente trabajo de titulación: **“Estudio Comparativo de Metodologías de Desarrollo de Software para aplicaciones móviles”** y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi trabajo de titulación en el Repositorio Institucional - Biblioteca Virtual.

Firma:

Cédula: 1150061032

Correo electrónico: andy.camacho@unl.edu.ec

Celular: 0967336276

Fecha: 17/09/2021

Carta de autorización por parte del autor para la Consulta de producción parcial o total, y publicación electrónica de texto completo.

Yo; **Andy Aron Camacho Herrera**, declaro se autor del Trabajo de Titulación: “**Estudio comparativo de metodologías de desarrollo de software para aplicaciones móviles**”, como requisito para obtener el título de Ingeniero en Sistemas; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y el exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de la autorización, firma el autor, en la ciudad de Loja, a cuatro días del mes de mayo del dos mil veintidós.

Firma:

Autor: Andy Aron Camacho Herrera

Cédula: 1150061032

Correo electrónico: andy.camacho@unl.edu.ec

Celular: 0967336276

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Edison Leonardo Coronel Romero Mg.Sc

Tribunal de Grado: Ing. Pablo Fernando Ordoñez Ordoñez Mg.Sc

Ing. Francisco Javier Álvarez Pineda Mg.Sc

Ing. Wilman Patricio Chamba Zaragocin Mg.Sc

Dedicatoria

Dedico este trabajo primeramente a Dios, quien me ha guiado por el buen camino y me ha dado vida y salud para cumplir con el sueño tan anhelado de ser ingeniero.

También dedico todo el esfuerzo y sacrificio puestos en este trabajo de titulación a mis padres que han sido el pilar fundamental para que yo logre mis objetivos y termine mis estudios de una manera satisfactoria. A mi madre Libia Camacho quien ha sido la que más se ha estado al pendiente de mi persona, desde que inicie mi carrera profesional dándome los recursos necesarios para facilitarme los estudios, y también darme el apoyo emocionalmente para enfrentar las adversidades suscitadas en el transcurso de mi formación profesional como ingeniero.

Dedicados a mis amigos Raquel Espinosa, Jonathan Tillaguango, Lorena Castillo y Evelyn Quevedo, quienes me han dado su apoyo y ánimos para culminar mi carrera y siempre han estado conmigo en toda esta etapa.

Andy Aron Camacho Herrera

Agradecimiento

Agradezco profundamente a nuestro Padre Celestial por todas las bendiciones derramadas en mi familia y amigos que de una u otra forma me han ayudado en el desarrollo del presente trabajo.

Le agradezco profundamente al Ingeniero Edison Leonardo Coronel Romero Mg. Sc., por compartir sus conocimientos y experiencias conmigo, ya que gracias a esto pude orientarme de la mejor manera a desarrollar mi Trabajo de Titulación dándome su apoyo de manera académica y emocional. También agradecerle por ayudarme a seleccionar el grupo experimental que fue de mucha importancia para obtener resultados comparativos.

También le agradezco a mi amiga y compañera Raquel Espinosa, quien ha sido una muy buena persona, y también por darme su apoyo incondicional en todos los ciclos cursados con ella. Igualmente, a mi amigo y compañero Jonathan Tillaguango quien fue un gran guía en toda mi carrera profesional, siempre dándome ánimo para salir adelante y poder llegar a culminar la carrera.

Finalmente, agradecerle a la Universidad Nacional de Loja, debido a que me brindó los recursos humanos que fueron de esenciales de principio a fin para el desarrollo del presente trabajo.

Índice de contenidos

Portada	i
Certificación del Trabajo de Titulación	ii
Autoría	iii
Carta de autorización.	iv
Dedicatoria	v
Agradecimiento	vi
Índice de contenidos	vii
Índice de tablas	ix
Índice de figuras	xi
Índice de anexos	xiii
1. Título	1
2. Resumen	2
2.1. Abstract	4
3. Introducción	5
4. Marco Teórico	7
4.1. Conceptos preliminares	7
¿Qué es una metodología?	7
Metodologías de desarrollo de software	7
Metodologías de software tradicionales	7
Metodologías ágiles	8
Ciclo de vida del software	9
4.2. Metodología RAD	10
4.3. Metodología RUP	11
4.4. Metodología en cascada	13
4.5. Prototipado	15
4.6. SCRUM	16
4.7. Desarrollo de aplicaciones móviles	20
4.8. Sistemas operativos para aplicaciones móviles	21
4.9. Sistema operativo Android de Google	21

4.10.	Sistema operativo iOS de Apple.....	22
4.11.	Trabajos relacionados.....	23
5.	Metodología.....	24
5.1.	Contexto.....	24
5.2.	Proceso.....	24
5.3.	Recursos.....	25
	Recursos científicos.....	25
	Recursos técnicos.....	26
	Recursos éticos.....	27
6.	Resultados.....	28
6.1.	Objetivo 1: Determinar las metodologías de desarrollo de software para aplicaciones móviles a evaluar y establecer los criterios de comparación.	28
6.2.	Objetivo 2: Desarrollar un entorno de experimentación para la evaluación de las metodologías de desarrollo de software para aplicaciones móviles.	46
6.3.	Objetivo 3: Evaluar las metodologías de desarrollo de software para aplicaciones móviles de acuerdo a los criterios de evaluación establecidos.....	69
7.	Discusión.....	82
7.1.	Desarrollo de la propuesta alternativa.....	82
7.2.	Valoración técnica, económica, ambiental.....	84
8.	Conclusiones.....	87
9.	Recomendaciones.....	89
10.	Bibliografía.....	91
11.	Anexos.....	94

Índice de tablas

Tabla 1. Roles estándar de Scrum	19
Tabla 2. Trabajos Relacionados	23
Tabla 3. Preguntas de Investigación de la RSL	29
Tabla 4. Cadenas de búsqueda.....	30
Tabla 5. Bases de datos científicas	31
Tabla 6. Criterios de inclusión	31
Tabla 7. Ejecución de las cadenas de búsqueda.....	32
Tabla 8. Estudios seleccionados	35
Tabla 9. Criterios por usabilidad	40
Tabla 10. Aplicabilidad	41
Tabla 11. Procesos y productos	42
Tabla 12. Capacidad	43
Tabla 13. Matriz comparativa	44
Tabla 14. Resumen de tareas	48
Tabla 15. Roles de Prototipado	49
Tabla 16. Requerimientos de la aplicación de prueba	50
Tabla 17. CU01-Registrar Usuario	52
Tabla 18. CU02-Identificar Usuario	53
Tabla 19. CU03-Realizar Pedido	53
Tabla 20. CU04-Visualizar Restaurantes.....	54
Tabla 21. CU05-Gestionar Cuenta	54
Tabla 22. CU06-Modificar Cuenta	55
Tabla 23. CU07- Inhabilitar Cuenta	55
Tabla 24. CU08- Recibir Notificación.....	56
Tabla 25. CU09-Gestionar Catálogo.....	57
Tabla 26. CU10-Agregar Menú.....	57

Tabla 27. CU11-Modificar Menú.....	58
Tabla 28. CU12-Eliminar Menú	59
Tabla 29. CU13-Ver promociones	62
Tabla 30. Reuniones-Prototipado	67
Tabla 31. Información de la experimentación	68
Tabla 32. Evaluación por categoría-Usabilidad	70
Tabla 33. Evaluación por categoría-Aplicabilidad	71
Tabla 34. Evaluación por categoría-Procesos y Productos.....	73
Tabla 35. Evaluación por categoría-Capacidad	73
Tabla 36. Comparativa Experimental.....	75
Tabla 37. Evaluación Experimental	77
Tabla 38. Resultados de la Evaluación por Categorías	79
Tabla 39. Niveles de puntuación	80
Tabla 40. Fortalezas y Debilidades de Prototipado.....	81
Tabla 41. Valoración Económica-Talento Humano.....	84
Tabla 42. Valoración Económica-Servicios.....	85
Tabla 43. Valoración Económica-Materiales de oficina.....	85
Tabla 44. Valoración Económica- Recursos de Hardware.....	85
Tabla 45. Valoración Económica- Recursos de Software	85
Tabla 46. Costo total del proyecto	86

Índice de figuras

Figura 1. Ciclo de vida de RUP [13].....	12
Figura 2. Etapas de la metodología en cascada.....	13
Figura 3. Características del ciclo de vida de modelo en cascada [15].....	14
Figura 4. Ciclo de vida de Prototipado.....	16
Figura 5. Ciclo de vida de Scrum [16].....	18
Figura 6. Sistemas operativos más usados en el 2020 [20].....	21
Figura 7. Proceso de la Revisión Sistemática de Literatura.....	29
Figura 8. Diagrama de Flujo del Proceso de selección de artículos primarios.....	33
Figura 9. Síntesis del proceso de análisis para los estudios finales.....	34
Figura 10. Metodologías de desarrollo de software móvil usadas durante la pandemia del covid-19.....	37
Figura 11. Criterios de elección de las metodologías de desarrollo de software móvil.....	38
Figura 12. Sistemas operativos para el desarrollo móvil.....	39
Figura 13. Proceso Experimental [26].....	46
Figura 14. Diagramas de Casos de Uso general.....	51
Figura 15. Diagrama de Caso de Uso-Gestionar Cuenta.....	51
Figura 16. Diagrama de Caso de Uso-Gestionar Catálogo.....	52
Figura 17. Prototipo v1.0-Registro y Autenticación.....	60
Figura 18. Prototipo V1.0-Menú principal.....	60
Figura 19. Prototipo v1.0-Realizar pedidos.....	61
Figura 20. Casos de Uso del Prototipo v2.0.....	61
Figura 21. Prototipo de la app de pedidos v2.0.....	63
Figura 22. Prototipo v2.0-menu desplegable para productos y promociones.....	63
Figura 23. Prototipo v2.0- Políticas, historial y perfil del restaurante.....	64
Figura 24. Base de Datos de la app de pedidos.....	64
Figura 25. Pruebas de escenarios Fase I.....	65
Figura 26. Pruebas escenarios Fase II.....	66

Figura 27. Fase II del proceso experimental.....	69
Figura 28. Resultados de la comparación experimental	79

Índice de anexos

Anexo 1: Revisión Sistemática de Literatura	94
Anexo 2: Pruebas de Escenarios	125
Anexo 3: Pruebas de Aceptación	129
Anexo 4: Prototipos de la aplicación.....	130
Anexo 5: Certificado de traducción del resumen.....	131

1. Título

Estudio comparativo de metodologías de desarrollo de software para aplicaciones móviles.

Comparative study of software development methodologies for mobile applications

2. Resumen

La pandemia del Covid-19 ha afectado varios mercados a nivel mundial y la industria del software no es la excepción, por lo que varias empresas han buscado nuevas formas de trabajo para mantener su negocio. Esta crisis ha traído consigo varias oportunidades, pero también muchos desafíos para los desarrolladores y proveedores de software en Latinoamérica. Como oportunidad, la demanda de software creció, especialmente en las aplicaciones móviles, debido a que varias empresas de varios mercados pasaron a ofrecer sus servicios por medio de este tipo de aplicaciones. El desafío que debían enfrentar los desarrolladores es el tratar de cumplir con esta demanda en poco tiempo, por lo que requerían elegir una metodología de desarrollo que se ajusten a estas medidas.

El objetivo del presente Trabajo de Titulación es realizar un estudio comparativo de metodologías de desarrollo de software para determinar qué metodología o etapas de la misma se acopla de mejor manera en el desarrollo de aplicaciones móviles. Con esta finalidad la pregunta de investigación es la siguiente: ¿Qué metodología de desarrollo de software se puede implementar en la creación de aplicaciones móviles en tiempos de pandemia del covid-19?, bajo este contexto, una correcta elección de una metodología de desarrollo de software permitirá la creación de aplicaciones móviles adaptándose a entornos turbulentos.

La pregunta de investigación se respondió mediante la ejecución de tres fases, en la primera fase, se efectuó una Revisión Sistemática de Literatura (RSL) que ayudó a determinar las metodologías de desarrollo de software para aplicaciones móviles más usadas durante la pandemia, dando como resultado a la metodología Prototipado. En esta misma fase se construyó una matriz de comparativa con 34 criterios de evaluación divididos en 4 categorías; esto con ayuda de una guía comparativa de metodologías de desarrollo de software y la RSL. En la segunda fase, se desarrolló un entorno de experimentación en un ámbito académico con ayuda de un grupo de estudiantes de noveno ciclo de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, el cual permitió evaluar en un ambiente real a la metodología seleccionada por medio del desarrollo de una aplicación móvil de prueba. En la tercera fase se realizó la evaluación donde se comprobó la hipótesis planteada relacionada con la pregunta de investigación y se determinó que Prototipado es una metodología de desarrollo de software que se adaptó al desarrollo de la aplicación móvil de prueba con un grado de satisfacción del 91%. También se logró determinar fortalezas y debilidades de Prototipado a partir de todo el proceso investigativo del presente TT. Finalmente, se presentan los trabajos futuros donde se propone realizar nuevas investigaciones y desarrollando un estudio comparativo de metodologías de desarrollo de software en un ámbito empresarial, esto con el fin de obtener resultados más sólidos al momento de elegir una metodología de desarrollo para aplicaciones móviles.

Palabras claves: app móvil, análisis, documentación, evaluación, metodología de desarrollo, Prototipado.

2.1. Abstract

The Covid-19 pandemic has affected many markets worldwide and the software industry is no exception, so several companies have sought new ways of working to maintain their business. This crisis has brought with it several opportunities, but also many challenges for software developers and suppliers in Latin America. As an opportunity, the demand for software grew, especially in mobile applications, due to several companies in various markets started to offer their services through this type of applications. The challenge that developers had to face was to try to meet this demand in a short time, so they needed to choose a development methodology that would fit these measures.

The objective of this Degree Project is to carry out a comparative study of software development methodologies to determine which methodology or stages of these methodologies is best suited for the development of mobile applications. With this purpose the research question is the following: Which software development methodology can be implemented in the creation of mobile applications in times of covid-19 pandemic? Under this context, a correct choice of a software development methodology will allow the creation of mobile applications adapting to turbulent environments.

The research question was answered through the execution of three phases, in the first phase, a Systematic Literature Review (SLR) was performed to help determine the most used software development methodologies for mobile applications during the pandemic, resulting in the Prototyping methodology. In this same phase, a comparative matrix was constructed with 34 evaluation criteria divided into four categories; this with the help of a comparative guide of software development methodologies and the SLR. In the second phase, an experimentation environment was developed in an academic environment with the help of a group of students belonging to ninth cycle of the Systems Engineering career of the Universidad Nacional de Loja, which allowed the evaluation of the selected methodology in a real environment through the development of a test mobile application. In the third phase, the evaluation was carried out, where the hypothesis related to the research question was tested and it was determined that Prototyping is a software development methodology that was adapted to the development of the mobile test application with a degree of satisfaction of 91%. It was also possible to determine strengths and weaknesses of Prototyping from the whole research process of this TT. Finally, future work is presented where it is proposed to conduct further research and developing a comparative study of software development methodologies in a business environment, in order to obtain more solid results when choosing a development methodology for mobile applications.

Keywords: mobile app, analysis, documentation, evaluation, development methodology, software, Prototyping.

3. Introducción

Las metodologías de desarrollo de software son consideradas como una base para el desarrollo y ejecución de cualquier proyecto de software, donde se considera que los miembros que conforman el equipo de desarrollo tengan la capacidad, disponibilidad y experiencia necesaria para cumplir a tiempo en el desarrollo de dichos proyectos [1].

Es muy importante saber elegir una metodología de desarrollo adecuada a las necesidades del cliente, además de analizar el tipo de producto de software a desarrollar, ya sean sistemas completos, aplicaciones web, aplicaciones de escritorio o las aplicaciones muy demandadas como son las móviles. Una metodología de desarrollo debe ser capaz de adaptarse a entornos estrictos o turbulentos; es decir que en caso de que suceda algún imprevisto el marco de trabajo del equipo no se vea afectado y se pueda continuar con el proyecto, minimizando los riesgos y el fracaso del equipo, como ejemplo se tiene la pandemia del Covid-19 donde afectó a muchas empresas de software debido a que no podían trabajar de forma presencial y tuvieron que analizar el panorama y determinar cómo se va a trabajar bajo esta nueva modalidad, pero también se presentaron nuevas oportunidades donde la mayoría de las empresas hicieron uso de la tecnología, especialmente a ofrecer sus servicios por medio de aplicaciones móviles, por tal motivo la demanda de estas aplicaciones creció, y fue un gran reto para los desarrolladores los cuales debían cumplir con la demanda. Ante esta situación, los desarrolladores tenían que elegir una metodología de desarrollo para aplicaciones móviles que les permitiera el desarrollo eficaz y eficiente de este tipo de aplicaciones y así poder mantener su empresa en el mercado durante esta crisis sanitaria.

El presente trabajo de titulación está encaminado a realizar un estudio comparativo de metodologías de desarrollo de software para aplicaciones móviles que ayude a determinar cuál metodología es mejor o cual se adapta de mejor manera a los requerimientos del cliente tomando en cuentas los recursos disponibles, el entorno de trabajo, la modalidad de trabajo y el tiempo establecido.

El Trabajo de Titulación fue realizado en un ámbito académico, durante el periodo de abril a septiembre del 2021 enfocado en un “Estudio comparativo de metodologías de desarrollo de software para aplicaciones móviles”, por lo que se requirió la participación de alumnos de noveno ciclo de la carrera de Ingeniería en Sistemas que cursaban en el periodo mencionado anteriormente, y fueron seleccionados en base a su disponibilidad y analizando su malla curricular en curso, esto se pudo dar gracias a la intervención del director del presente Trabajo de Titulación.

La pregunta de investigación del presente TT es la siguiente: ¿Qué metodología de desarrollo de software se puede implementar en la creación de aplicaciones móviles en tiempos de pandemia del covid-19?, por ello se planteó como objetivo principal “Realizar un estudio

comparativo de metodologías de desarrollo de software para determinar qué metodología o etapas de la misma se acopla de mejor manera en el desarrollo de aplicaciones móviles”, por ende, se definieron tres objetivos específicos, siendo el primero “Determinar las metodologías de desarrollo de software para aplicaciones móviles”, el segundo “Desarrollar un entorno de experimentación para la evaluación de las metodologías de desarrollo de software para aplicaciones móviles”, y el tercero “Evaluar las metodologías de desarrollo de software para aplicaciones móviles de acuerdo a los criterios de evaluación establecidos”.

Para dar cumplimiento con los objetivos planteados se realizó una Revisión Sistemática de Literatura que ayudo a determinar a Prototipado como la metodología más usada durante la pandemia del Covid-19, además ayudó a la construcción de la matriz de evaluación en conjunto con una guía comparativa de metodologías de desarrollo de software [2]. Luego se desarrolló un entorno de experimentación con el grupo de estudiantes seleccionados el cual desarrolló una aplicación móvil de prueba aplicando la metodología asignada, finalmente se realizó la comparativa por medio de la matriz creada para cumplir el tercer objetivo.

A continuación, se describe de manera general las secciones del presente Trabajo de Titulación. En la sección Revisión de Literatura se elaboran 14 capítulos del área de estudio que sirvieron como base fundamental para la ejecución del presente TT. La sección de Materiales y Métodos, se describe el contexto y procedimiento para el desarrollo del TT, además se indica los recursos utilizados y los participantes que intervinieron en el desarrollo. En la sección de Resultados, se indica la evidencia que se obtuvo durante la ejecución de TT, dividida de acuerdo a los tres objetivos específicos. En la sección de Discusión, se hace un análisis de los resultados obtenidos desde el punto de vista del autor basado en evidencias, también se describe como se cumple cada objetivo. La sección de Conclusiones, describe los sucesos más relevantes que se encontraron durante el desarrollo del TT. Finalmente, en la sección de Recomendaciones, se indican aspectos a considerar para los lectores y posibles trabajos futuros a desarrollar.

4. Marco Teórico

A continuación, se presentan las bases teóricas que sustentan el presente Trabajo de Titulación (TT), dicha información ha sido obtenida a través de un proceso de revisión bibliográfica. Esta sección inicia con los conceptos más relevantes entorno a la temática, y posteriormente, se presenta una tabla con todos los estudios que han sido seleccionados durante el proceso de revisión bibliográfica.

4.1. Conceptos preliminares

¿Qué es una metodología?

Según un estudio de Andrea Gordillo [3] indica que: “Una metodología es el conjunto de operaciones o actividades que, dentro de un proceso pre establecido, se realizan de manera sistemática para conocer y actuar sobre la realidad social”.

La definición de metodología puede ser entendida como una disciplina cuya función es la elaboración, definición y sistematización de una serie o conjunto de técnicas, métodos y procedimientos, los cuales cumplen un papel muy importante en la elaboración de un proyecto.

La función de una metodología, se encamina en examinar, valora, refutar o corroborar la eficacia de los métodos en los diversos campos del conocimiento. Si estos métodos en cuanto a diseño y sentido no cumplen con su eficacia, significa que hay errores de concepción y contenido por cuanto a su sustentación. Esto implica que tienen límites cognitivos y empíricos, lo cual impide que sean confiables para el trabajo de investigación [4].

Metodologías de desarrollo de software

Según Oscar Tinco en su artículo sobre un estudio de metodologías de desarrollo de software [5] nos dice que:” Una metodología de desarrollo de software es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Estas clases de metodologías están formadas por fases, cada una de las cuales se puede dividir en sub-fases, con el objetivo de guiar al desarrollador de sistemas a elegir las técnicas más apropiadas durante el desarrollo del proyecto, así mismo ayudará a planificarlo, gestionarlo, controlarlo y evaluarlo”.

Metodologías de software tradicionales

Estas metodologías imponen una disciplina de trabajo sobre el proceso de desarrollo de software, para ello, se hace énfasis en la planificación total de todo el trabajo a realizar, una

vez que está todo detallado, se comienza el ciclo de desarrollo del producto de software.

Un estudio desarrollado por Roberth Figueroa [6] da a entender que las metodologías tradicionales se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y por supuesto una documentación detallada, pero no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados valga la redundancia, cuando se trabaja en un entorno donde los requisitos no pueden predecirse o bien pueden variar.

Un método tradicional utiliza un enfoque lineal donde las etapas del proceso de desarrollo deben completarse en un orden secuencial. Esto significa que una etapa debe completar antes de que comience la siguiente. Estas etapas generalmente comprenden lo siguiente:

- Recopilación de requisitos y documentación
- Diseño de sistemas
- Codificar y hacer pruebas unitarias
- Pruebas de sistema
- Pruebas de aceptación del usuario
- Corrección de errores
- Entrega del producto

En el mundo del desarrollo de software las empresas o instituciones educativas han optado por aplicar una o varias metodologías tradicionales para llevar a cabo sus proyectos.

Entre las metodologías y modelos tradicionales más conocidas tenemos las siguientes:

- RUP
- Modelo en Cascada
- Prototipado
- Espiral
- RAD
- Modelo Evolutivo

Metodologías ágiles

En los últimos años estas metodologías han sido muy populares, surgen como respuesta a ciertos problemas que se presentan al momento de utilizar una metodología tradicional, y se basa en dos aspectos puntuales: el retrasar las decisiones y la planificación adaptativa, permitiendo potenciar aún más el desarrollo de software a gran escala [6].

La metodología ágiles utilizan un enfoque iterativo y basado en equipos. Su principal objetivo es entregar rápidamente la aplicación con componentes completos y funcionales. En lugar de completar las tareas de desarrollo de software en secuencia, se completan en sprints que se

ejecutan desde alrededor de 1 a 4 semanas, y donde se completa una lista de entregas en cada sprint [6]. Las tareas que no se completan dentro del sprint se vuelven a priorizar y se incluyen en los sprints futuros. Esto significa que las diferentes etapas del ciclo de vida del desarrollo del software pueden revisarse según sea necesario.

Un enfoque ágil típico implica las siguientes etapas:

- Iniciación del proyecto
- Sprint planificado
- Producción

Con el método tradicional, los detalles de todo el proyecto se han visualizado y definido antes de que comience el proyecto. Por el contrario, la metodología ágil permite una mayor flexibilidad, debido a que los cambios pueden realizarse más fácilmente incluso después de que se inicia el proyecto [7]. Se emplea mejor si el alcance del proyecto no puede definirse claramente con antelación. Esto también significa que realizar cambios durante el desarrollo de software no planificados con el método tradicional, es más costoso que con ágil.

Toda metodología ágil se respalda en los cuatro valores señalados en el manifiesto ágil [8], como ya se ha dicho anteriormente fue diseñado en respuestas a las nuevas exigencias y marcan la diferencia con los sistemas tradicionales:

1. Los individuos y su interacción, por encima de los procesos y las herramientas.
2. El software que funciona, frente a la documentación exhaustiva
3. La colaboración con el cliente, por encima de la negociación contractual.
4. La respuesta al cambio, por encima del seguimiento de un plan.

Las metodologías ágiles se han ganado un puesto en el ámbito empresarial para desarrollar proyectos con baja y alta complejidad. Entre las metodologías ágiles más populares están:

- Extreme Programming (XP)
- Scrum
- Kanban
- Design Sprint
- Iconix
- Agile Inception

Ciclo de vida del software

Las fases, procesos o ciclos de vida del desarrollo de software contemplan una serie de pasos a fases necesarias para validar el desarrollo de software y así garantizar que este cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo, asegurándose de que los métodos usados son apropiados [9].

Según la ISO/ICE/IEEE 12207:2017 referente al ciclo de vida del software establece que: “Un marco común para los procesos del ciclo de vida de los programas informáticos, con una terminología bien definida, a la que pueda remitirse la industria del software. Contiene procesos, actividades y tareas aplicables durante la adquisición, el suministro, el desarrollo, el funcionamiento, el mantenimiento o la eliminación de sistemas, productos y servicios informáticos. Estos procesos del ciclo de vida se llevan a cabo mediante la participación de los interesados, con el objetivo final de lograr la satisfacción del cliente” [10].

Como se ha dicho anteriormente el ciclo de vida de desarrollo de software está compuesto de fases o etapas, estas son:

- **Planificación:** Aquí se planifican las tareas que se van a realizar durante el proyecto con el fin de que este se cumpla a cabalidad y en el tiempo establecido.
- **Análisis:** Se debe analizar los requisitos del cliente y las funciones que el software realizará.
- **Diseño:** De acuerdo al análisis se estudian las posibles opciones de implementación para construir el software.
- **Codificación:** Se eligen las herramientas adecuadas y el lenguaje de programación apropiado al software que se va a desarrollar. Esta fase depende del diseño y del entorno en que se va a construir el software.
- **Pruebas:** Esta fase tiene como objetivo identificar los posibles errores que no se han podido detectar en las fases anteriores y tratar de mitigarlos, con el fin de que el software sea de calidad y fil a los requisitos del cliente.
- **Despliegue:** En esta fase se debe poner el software en funcionamiento, teniendo en cuenta las dependencias existentes entre los diferentes componentes del mismo.
- **Uso y mantenimiento:** Esta es la etapa final donde los usuarios finales hacen uso del software y se debe estar dispuesto a mantenerlo y mejorarlo, lo cual implica eliminar posibles errores, adaptarlos a nuevas necesidades y añadirle nuevas funcionalidades si el cliente lo pide.

4.2. Metodología RAD

La metodología RAD (Rapid Application Development) o DRA (Desarrollo Rápido de Aplicaciones), es una metodología de desarrollo de software donde abarca el desarrollo interactivo, la creación de prototipos y el uso constante de las herramientas CASE.

En un estudio de Robison Campaña [11] nos indica que para aplicar la metodología CASE se debe tener en cuenta 4 componentes: Personas, herramientas, metodología y gestión. Su filosofía es entregar sistemas de alta calidad, en poco tiempo y con un coste bajo de inversión, lo cual se debe seguir ciertas pautas para llevar una correcta aplicación de la misma.

Como todas metodologías tiene sus fases, estas son:

- **Planeación de requerimientos:** Se definen las funciones del negocio, describe las características del software, las áreas de influencia del software y al alcance del mismo.
- **Diseño funcional:** La modelación de datos y procesos se lleva a través de talleres construyendo paralelamente prototipos funcionales que se van probando, detallando, perfeccionando con el mismo usuario, siempre ayudados por las herramientas CASE [12].
- **Construcción o desarrollo:** En esta fase se completa el desarrollo de sistemas, las iteraciones van arrojando componentes y se realiza pruebas de integración de los mismos, de acuerdo a los planes de trabajos establecidos.
- **Implementación:** Aquí el usuario final es el eje del modelo, por lo tanto, él será el que prueba el software y proporciona su conformidad. Luego con la herramienta CASE correctamente configurada se implementa el software.

Entre los roles de esta metodología tenemos los siguientes:

- **Sponsor:** Es representado por un ejecutivo de alto nivel de la organización y que espera resultados en el menor tiempo posible.
- **Equipos de planeación de requerimientos:** Es representado por usuarios con experiencia tanto en la parte de procesos de negocio como analistas de información o afines.
- **Diseñadores:** Este equipo realiza el diseño, modelos y demás documentos técnicos de ingeniería.
- **Administrador del proyecto:** Controla y administra la ejecución del proyecto.
- **Equipo de construcción:** Según la metodología recomienda de dos a seis personas, quienes deben tener amplia experiencia en el manejo de operaciones de herramientas CASE.

4.3. Metodología RUP

La metodología RUP (RATIONAL UNIFIED PROCESS), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Su objetivo o meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos [13].

RUP tiene dos dimensiones:

- El eje horizontal representa tiempo y demuestra los aspectos de ciclo de vida del proceso
- El eje vertical representa las disciplinas que agrupan actividades definidas lógicamente por la naturaleza.

El ciclo de vida de esta metodología es una implementación del desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semiordenadas. Este ciclo organiza las tareas en fases e iteraciones [13]. RUP divide el proceso en cuatro fases, las cuales son grandes y formales iteraciones en un número variable según el proyecto, estas fases se muestran en la Figura 1.

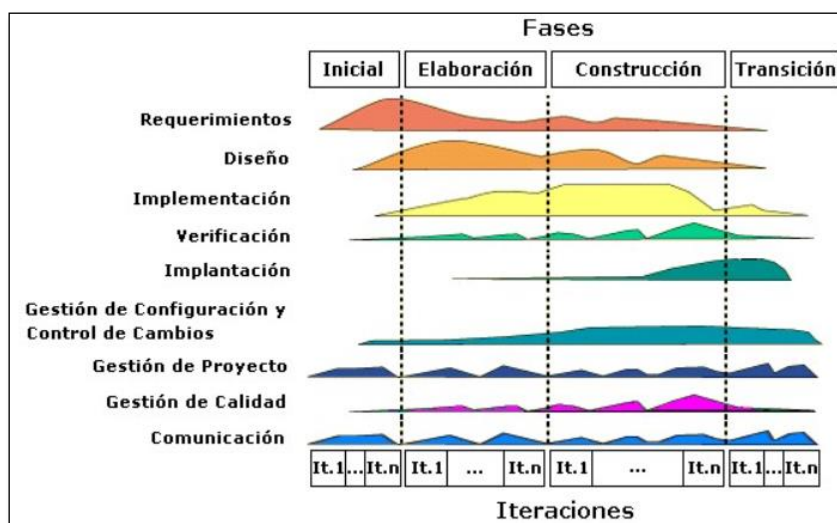


Figura 1. Ciclo de vida de RUP [13].

- **Fase inicial:** Se define el ámbito y los objetivos de proyecto, además la funcionalidad y capacidades del producto.
- **Elaboración:** Se estudia el problema a profundidad para lo cual se define una arquitectura básica y se planifica el proyecto considerando los recursos disponibles.
- **Construcción:** El proceso se lleva a través de iteraciones donde cada iteración involucra tareas de análisis, diseño e implementación.
- **Transición:** Se libera el producto y se entrega al usuario final. Aquí se incluyen tareas de marketing, empaquetado atractivo, instalación, configuración, entrenamiento, soporte, mantenimiento.

4.4. Metodología en cascada

Es un método que consiste en desarrollar un proyecto de forma secuencial. Requiere un alto detalle de los requerimientos para empezar [14]. Al contrario que en los modelos iterativos, cada una de estas fases se ejecuta tan solo una vez. Los resultados de cada una de las fases sirven como hipótesis de partida para la siguiente.

Como se puede observar en la Figura 2 el modelo en cascada está compuesto por cinco fases: requisitos, diseño, implementación, verificación, mantenimiento. Como se decía anteriormente, sigue una serie de etapas de forma sucesiva, la etapa siguiente empieza cuando termina la etapa anterior.

Sus fases serán detalladas a continuación:

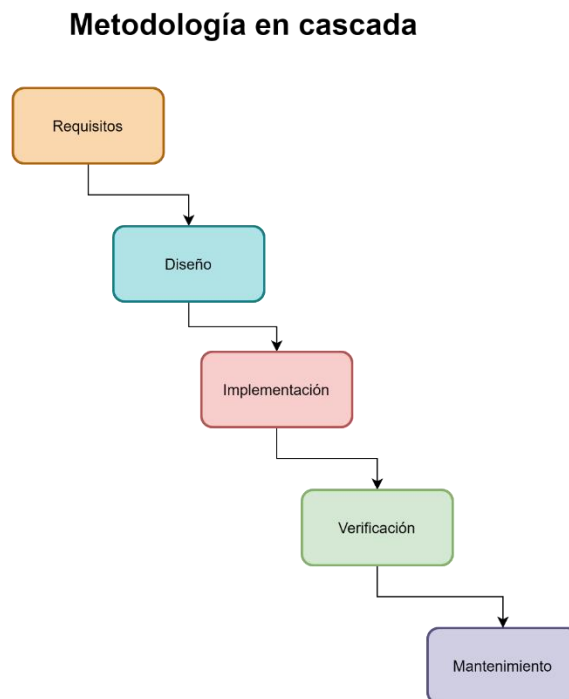


Figura 2. Etapas de la metodología en cascada.

- **Requisitos del software:** Se hace un análisis de las necesidades del cliente para determinar las características del software a desarrollar, y se especifica todo lo que se debe hacer en el sistema sin entrar en detalles técnicos. Se debe ser cuidadoso en esta primera fase, debido a que en este modelo no se pueden añadir nuevos requisitos en mitas del proceso de desarrollo.
- **Diseño:** Se describe la estructura interna del software, y las relaciones entre las entidades que lo componen. Aquí se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipos.

- Implementación: Se programan los requisitos especificados haciendo uso de las estructuras de datos diseñados en la fase anterior. La programación es el proceso que lleva de la formulación de un problema de computación a un programa que se ejecute produciendo los pasos necesarios para resolver dicho problema.
- Verificación: Una vez terminada la fase de implementación se verifica que todos los componentes del sistema funcionen correctamente y cumplen con los requisitos.
- Instalación o mantenimiento: Después de que se han desarrollado todas las funcionalidades del software y se ha comprobado que funcionen correctamente, se inicia con la instalación y mantenimiento. Se instala la aplicación en el sistema y se comprueba que funcionen correctamente en el entorno en que se va a utilizar. A partir de este punto, se debe asegurar de que el software funcione y hay que destinar recursos a mantenerlos. El mantenimiento del software consiste en la modificación del producto después de haber entregado al cliente, ya sea para corregir errores o para mejorar el rendimiento o las características.

Un estudio realizado por Demián Gutiérrez [15] nos simplifica estas fases y sus peculiares características como se puede observar en la Figura 3.

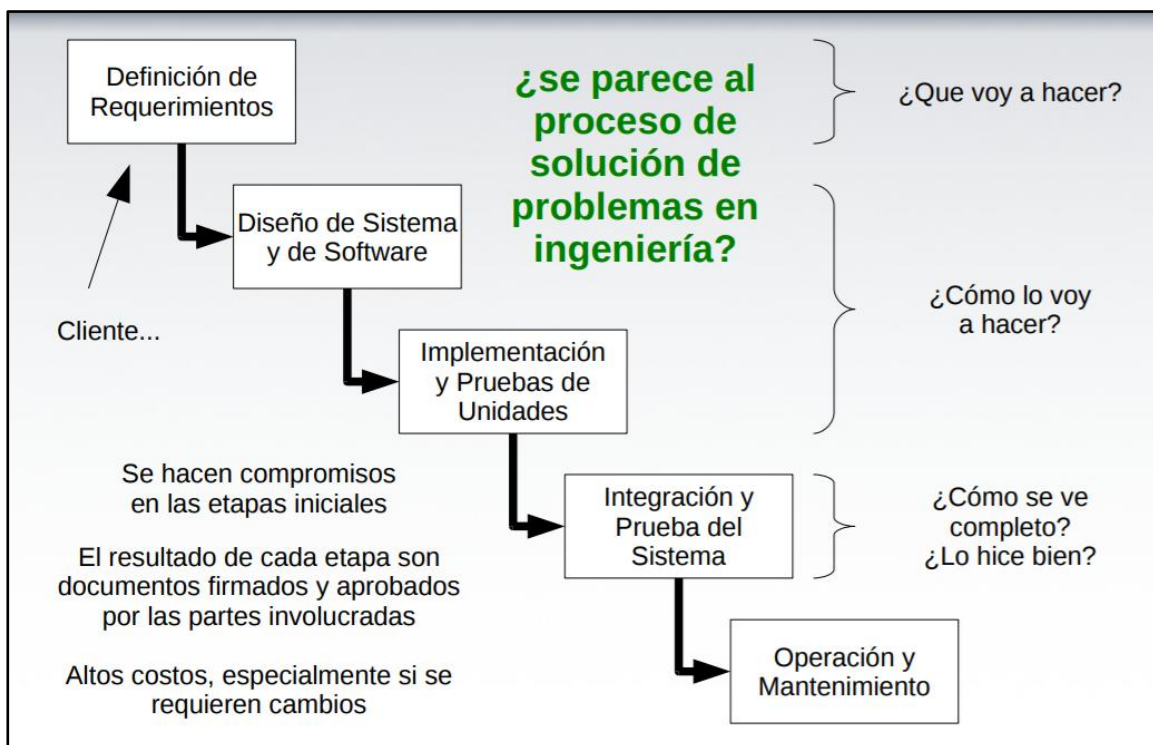


Figura 3. Características del ciclo de vida de modelo en cascada [15]

4.5. Prototipado

Un prototipo es una versión preliminar de un sistema de información con fines de demostración o evaluación.

El modelo de prototipos en Ingeniería de software, pertenece a los modelos de desarrollo evolutivo. El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos [1].

El prototipo de requerimientos, es la creación de una implementación parcial de un sistema, para el propósito explícito de aprender sobre los requerimientos del sistema. Esto es dado a los usuarios, cliente o representantes de ellos, posibilitando que ellos experimenten con el prototipo. Estos individuos luego proveen la retroalimentación sobre la aplicación construida, si el prototipo que se les presentó cumple con sus expectativas o no. El prototipo puede ser usado como parte de la fase de requerimientos o justo antes de la fase de requerimientos. Por otra parte, el prototipo puede servir inmediatamente antes de algún o todo el desarrollo incremental en modelos incremental o evolutivo.

Prototyping o prototipado ha sido muy usado muy frecuentemente desde la década de los 60's, debido a la especificación de requerimientos para sistemas complejos tiende a ser relativamente dificultoso de cursar [1]. Muchos usuarios y clientes encuentran que es mucho más fácil proveer retroalimentación convenientemente basada en a la manipulación, leer una especificación de requerimientos potencialmente ambigua y extensa.

La construcción de software por prototipos tiene algunas características que se deben tomar en cuenta al momento de desarrollar una aplicación, entre estas están:

- Es un método menos formal de desarrollo.
- El prototipo es una técnica para comprender las especificaciones.
- El prototipo puede ser eliminado
- Un prototipo puede llegar a ser parte del producto final.

El ciclo vida de prototipado consta de ciclos iterativos e incrementales los cuales constan de 6 fases como se puede observar en la Figura 4. Ciclo de vida de PrototipadoFigura 4.

Construcción de software por prototipos

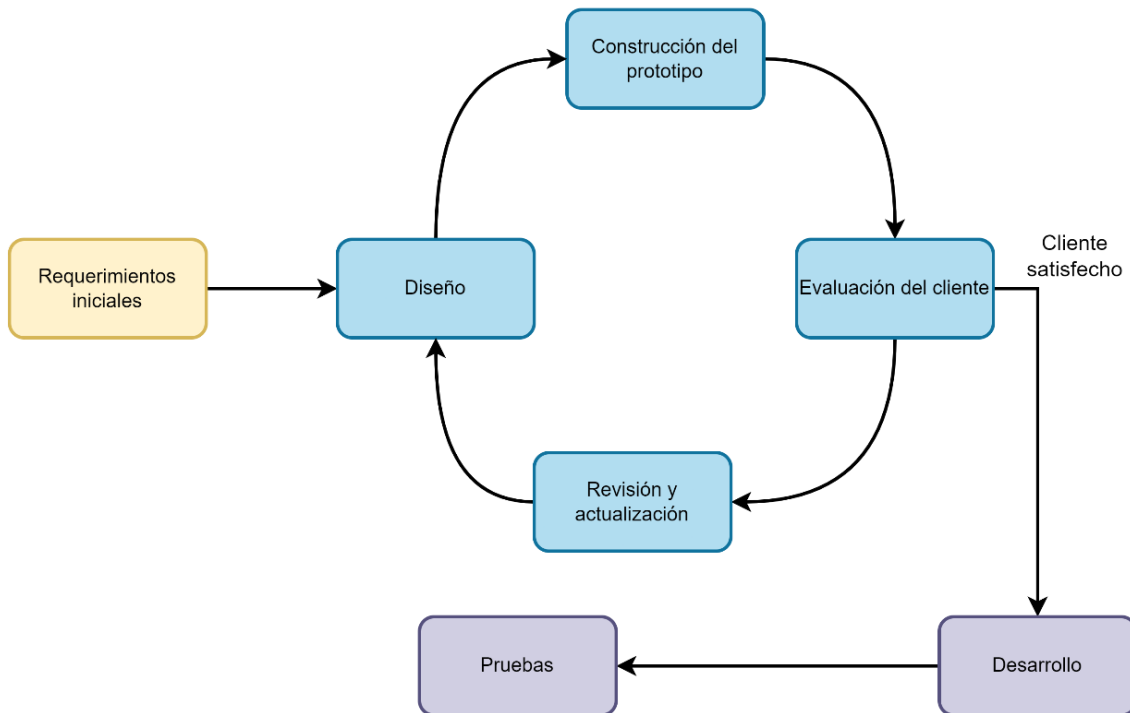


Figura 4. Ciclo de vida de Prototipado

4.6. SCRUM

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software, su principal objetivo es maximizar el retorno de la inversión para su empresa o equipo de desarrollo. Se basa en construir la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, autogestión e innovación [16].

El concepto de Scrum tiene sus orígenes sobre los nuevos procesos de desarrollo utilizados en productos exitosos en Japón y los Estados Unidos. Los equipos que desarrollaron estos productos partían de requisitos muy generales, así como novedosos, y debían salir al mercado en mucho menos tiempo del que se tardó en lanzar productos anteriores, estos equipos seguían patrones de ejecución de proyecto muy similares. De acuerdo a la historia [1], en este estudio se comparaba la forma de trabajo de estos equipos altamente productivos y multidisciplinarios con la colaboración entre los jugadores del Rugby y su formación de Scrum, de la cual se tomó su nombre.

Fundamentos de Scrum

Scrum se basa en los siguientes puntos:

- El desarrollo incremental de los requisitos del proyecto en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita). Las

iteraciones se pueden entender como mini proyectos, en todas las iteraciones se repite un proceso de trabajo similar para proporcionar un resultado completo sobre el producto final, de manera que el cliente pueda obtener los beneficios del proyecto de forma incremental. Para ellos, cada requisito se debe completar en una única iteración. El equipo debe realizar todas las tareas necesarias para completarlo (incluyendo pruebas y documentación) y que esté preparado para ser entregado al cliente con el mínimo esfuerzo necesario. De esta manera no se deja para el final del proyecto ninguna actividad arriesgada relacionada con la entrega de requisitos.

- La priorización de los requisitos por valor para el cliente y coste de desarrollo en cada iteración. Para que un proyecto proporcione el mejor resultado posible, y como soporte fundamental al control empírico del proyecto, ese necesario repriorizar los requisitos de manera regular, en cada iteración, según el valor que proporcionan al cliente en ese momento y su coste estimado de desarrollo. Como resultado de esta repriorización se actualiza la lista de requisitos priorizada (Producto Backlog).
- El control empírico del proyecto. Por un lado, al final de cada iteración se demuestra al cliente el resultado real obtenido, de manera que pueda tomar las decisiones necesarias en función de lo que se observa y del contexto del proyecto en ese momento. Por otro lado, el equipo se sincroniza diariamente y realiza las adaptaciones necesarias.
- La potenciación del equipo, que se compromete a entregar unos requisitos y para ello se le otorga la autoridad necesaria para organizar su trabajo.
- La sistematización de la colaboración y la comunicación tanto entre el equipo y como el cliente.
- El timeboxing de las actividades del proyecto, para ayudar a la toma de decisiones y conseguir resultados. La técnica del timebox consiste en fijar el tiempo máximo para conseguir ciertos objetivos, tomar una decisión o realizar unas tareas, y hacer lo mejor que podamos en ese intervalo. De esta manera, en lugar de ponerse a trabajar en algo hasta que esté hecho, de antemano se acuerda que sólo se dedica un tiempo limitado. La conciencia de esta limitación temporal favorece la priorización de objetivos/tareas y fuerza la toma de decisiones.

Ciclo de vida de Scrum

La ejecución de un proyecto en Scrum, es mediante bloques temporales cortos y fijos (iteraciones de 2 a 4 semanas máximo). Cada iteración tiene que proporcionar un resultado completo, un incremento del producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

Todo comienza con una idea de lo que se quiere construir, se realiza una reunión entre las partes interesadas donde se recolectan estas ideas y requerimientos de los Stakeholders o personas a las que les desarrollamos el producto. A este conjunto de requerimientos ordenados por los que aportan más valor se llama Product Backlog.

Luego se realiza el primero de los Sprints con la planificación de Scrum (Scrum Planning), posteriormente se hacen las reuniones diarias de sincronización del equipo (Daily Scrum), la revisión del sprint (Sprint Review), el ciclo termina con una reunión de retrospectiva (Sprint Retrospective) que tiene por objetivo la mejora constante [16].

En este punto se puede dar inicio al siguiente Sprint de duración fija entre una y cuatro semanas, donde el equipo de desarrollo comienza a trabajar sobre Product Backlog. Todo esto se repite hasta que el proyecto se considere completado.

En la Figura 5, se puede observar de manera gráfica el ciclo de vida de Scrum e interacción entre todos sus componentes.

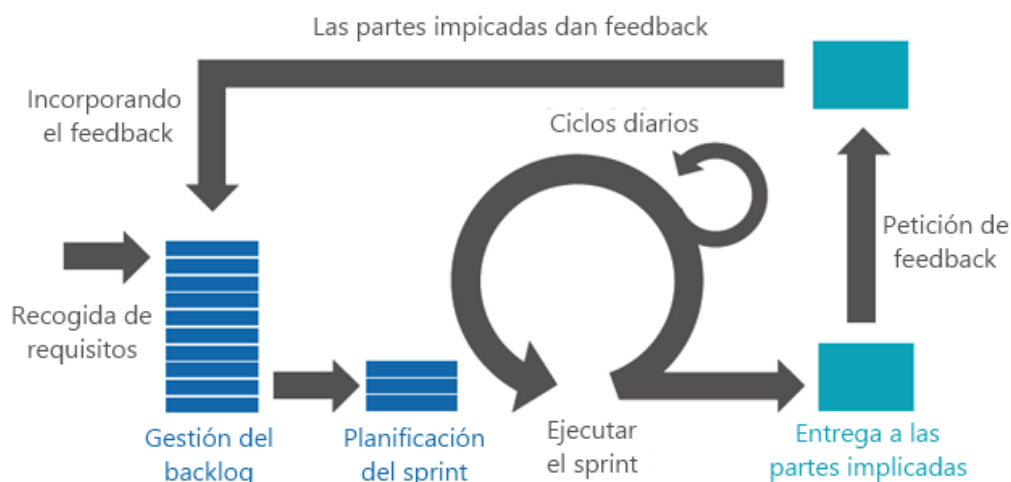


Figura 5. Ciclo de vida de Scrum [16].

Roles de Scrum

Las personas que intervienen, o tienen relación directa o indirecta con el proyecto, se clasifican en dos grupos: comprometidos e implicados [17]. Esto es interesante, debido a que a los comprometidos se los llama “cerdos”, y a los implicados “gallinas”; es una metáfora que ilustra de forma gráfica la diferencia entre “compromiso” e “implicación” en el desarrollo de un proyecto. La metáfora se la conoce de la siguiente manera:

Una gallina y un cerdo paseaban por la carretera. La gallina preguntó al cerdo: “¿Quieres abrir un restaurant conmigo?”.

El cerdo consideró la propuesta y respondió: “Si, me gustaría. ¿Y cómo lo llamaríamos?”.

La gallina respondió: “Huevos con jamón”.

El cerdo se detuvo, hizo una pausa y contestó: “Pensándolo mejor, creo que no voy a abrir un

restaurant contigo. Yo estaría realmente comprometido, mientras que tu estarías sólo implicada” [17].

Esto hace referencia a dos tipos de personas que participan en el desarrollo de un proyecto, los que en verdad ponen todo su esfuerzo y compromiso para que el proyecto surja, y los que tienen otros intereses y solo contarían como implicados. En Tabla 1, se ilustra de mejor manera estos roles.

Tabla 1.
Roles estándar de Scrum

COMPROMETIDOS	IMPLICADOS
Scrum Master	Otros interesados (dirección, gerencias, comerciales, marketing, etc.)
Dueño del producto (Product Owner)	
Equipo de desarrollo	

- **Propietario del producto:** Es la persona responsable de lograr el mayor valor de producto para los clientes, usuarios y resto de implicados.
- **Equipo de desarrollo:** Es un grupo de trabajo que desarrollan el producto.

Scrum Master

Es el responsable del cumplimiento de las reglas del marco de scrum; asegura que éstas sean comprendidas por la organización y de que se trabaje conforme a ellas. El Scrum Master asesora y da la formación necesaria al propietario del producto y al equipo, configura, diseña y mejora de forma continua las prácticas ágiles de la organización. Su objetivo principal es que tanto el equipo como el cliente sean capaces de organizarse y trabajar con autonomía [18].

Otra responsabilidad del Scrum Master es moderar las reuniones de scrum diarias, gestionar las dificultades de dinámica de grupo que puedan surgir en el equipo, y solucionar los impedimentos detectados durante el scrum diario para que el sprint siga avanzando.

Producto Owner o Dueño del Producto

Es quien toma las decisiones del cliente, su responsabilidad es el valor del producto.

Según Marta Palacios en su artículo “Scrum Manager Temario Troncal 1” [18], manifiesta que para simplificar la comunicación y toma de decisiones es necesario que este rol recaiga en una única persona. Si el cliente es una organización grande, o con varios departamentos, puede adoptar la forma de comunicación interna que consideren oportuna, pero en el equipo de trabajo solo se integra a una persona. Esta representa al cliente y debe tener el

conocimiento y las atribuciones necesarias para tomar las decisiones que les corresponden. Según las circunstancias del proyecto es posible incluso que el propietario del producto delegue en el equipo, o en alguien de su confianza, pero incluso en esos casos mantiene sus responsabilidades, estas son:

- Desarrollo y administración de la pila del producto
- Exposición de la visión e historias de usuario, y participación en la reunión de planificación de cada sprint.

Equipo de desarrollo

El equipo de desarrollo está conformado por un grupo de profesionales que realizan el incremento de cada sprint.

Se recomienda que un equipo de Scrum tenga entre 3 y 8 personas, pero esto depende de la dimensión del proyecto y la disponibilidad de los recursos destinados al mismo.

No sólo se trata de un grupo de trabajo formado por un arquitecto, diseñador o analista, programadores y testers; es un equipo multifuncional, en el que todos los miembros trabajan de forma solidaria con responsabilidad compartida. Es posible que algunos miembros sean especialistas en áreas concretas, pero la responsabilidad es el incremento de cada sprint y recae sobre el equipo de desarrollo en conjunto.

Un equipo de Scrum responde en su conjunto donde trabaja de forma cohesionada y autoorganizada. No hay un gestor para delimitar, asignar y coordinar las tareas. Son los propios miembros los que lo realizan.

En un equipo de Scrum:

- Todos conocen y comprenden la visión del propietario del producto.
- Aportan y colaboran con el propietario del producto en el desarrollo de la pila del producto.
- Comparten de forma conjunta el objetivo de cada sprint y la responsabilidad del logro.
- Todos los miembros participan en las decisiones.
- Se respetan las opiniones y aportes de todos.
- Todos conocen el modelo de trabajo con Scrum.

4.7. Desarrollo de aplicaciones móviles

El desarrollo de aplicaciones móviles ha crecido exponencialmente en los últimos 10 años, esto se debe al buen desempeño que tienen estos dispositivos y al uso excesivo de internet. Por tal motivo existe una amplia gama de aplicaciones en el mercado las cuales facilitan muchas tareas cotidianas donde se las puede resolver desde cualquier lugar solo usando este pequeño dispositivo. El desarrollo de software para las plataformas móviles viene con

características únicas y restricciones que se aplican a la mayoría de las etapas del ciclo de vida [19]. Las características distintivas más importantes son: un alto nivel competitividad, corto tiempo de entrega, movilidad, portabilidad, capacidades específicas y contantemente cambiantes de las terminales, sistemas operativos diferentes e incompatibles, entre otras.

4.8. Sistemas operativos para aplicaciones móviles

Como se puede observar en la Figura 6, los principales sistemas operativos que lideran en el mercado son Android y iOS. Según el StatCounter Global Stats 2020, Android es el más usado, con un 71.42%, seguido por iOS con un 27.95% [20].

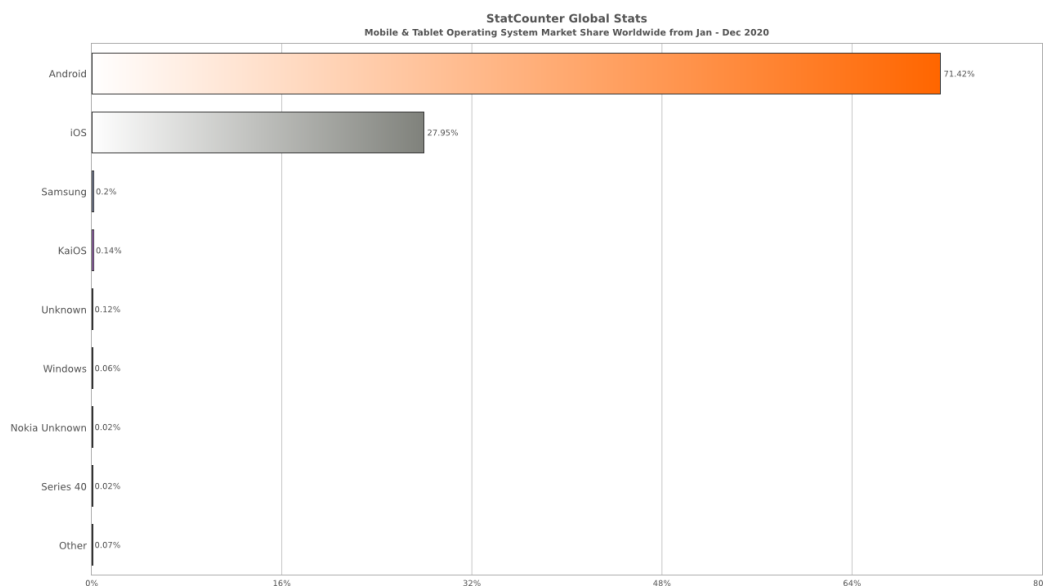


Figura 6. Sistemas operativos más usados en el 2020 [20].

4.9. Sistema operativo Android de Google

Android es un sistema operativo móvil diseñado para dispositivos móviles con pantalla táctil como teléfonos inteligentes o tablets, pero que también los encontramos en otros dispositivos como relojes inteligentes, televisores o incluso en los sistemas multimedia de algunos modelos de coches [19]. Este sistema operativo fue desarrollado por Google y basado en el Kernel de Linux y otros softwares de código abierto, y que se ha convertido en el principal responsable de la popularización de muchos dispositivos inteligentes por el hecho de facilitar el uso de una gran cantidad de aplicaciones de forma sencilla.

Desde su lanzamiento hasta el día de hoy, Android ha recibido numerosas actualizaciones. Diferentes versiones del sistema que han ido arreglando fallos detectados, añadiendo nuevas funciones, soportes para nuevas tecnologías, etc.

4.10. Sistema operativo iOS de Apple

iOS es un sistema operativo lanzado y utilizado por Apple. Su nombre proviene de iPhone OS, es decir, iPhone Operative System o Sistema Operativo de iPhone. Utiliza las siglas, iOS [21]. Se lanzó originalmente para el teléfono de la marca, aunque también se ha utilizado durante años en otros dispositivos de la compañía como en algunos de los reproductores de música iPod o en las tabletas iPad.

Según un estudio de Gabriel Pedrozo [21] indica que iOS es un sistema cerrado y exclusivo debido a que no puede utilizarse en otros dispositivos, sólo los que son de la marca Apple. Esta es una gran diferencia con respecto a Android, debido a que este último puede utilizar en una infinidad de dispositivos de todas las marcas. Además, iOS también permite instalar una serie de aplicaciones de las que vienen por defecto, con el objetivo de brindar calidad de servicio al igual que su competencia directa.

4.11. Trabajos relacionados

De acuerdo a la revisión bibliográfica realizada en la Tabla 2, se presentan los trabajos o estudios relacionados con el presente TT.

Tabla 2.
Trabajos Relacionados

Trabajo relacionado	Ref.	Resumen
Estudio Comparativo de Metodologías de Desarrollo Ágil en Base al Desarrollo de una Aplicación Móvil. (Universidad Central del Ecuador)	[22]	Este trabajo de titulación aborda y explora la eficiencia y eficacia del uso de las metodologías ágiles en la construcción de aplicaciones móviles, donde se realiza la experimentación creando un prototipo de una aplicación móvil destinado a localización de mascotas.
Metodología de desarrollo ágil para sistemas móviles: Introducción al desarrollo con Android y el iPhone	[23]	En este trabajo de doctorado se realiza una comparativa de metodologías de desarrollos ágiles para crear aplicaciones móviles, de igual manera realiza un caso práctico creando una aplicación tanto para dispositivos son sistemas operativos Android y para iOS
Modelo de Evaluación de Metodologías para el Desarrollo de Software	[24]	Este trabajo de postgrado diseña un modelo de evaluación de metodologías de desarrollo de software con el fin de ayudar en la selección de una correcta metodología para la una buena gestión de proyectos y éxito en el producto final.
Criterios de selección de metodologías de desarrollo de software	[5]	En este artículo de revista desarrolla una comparativa entre metodologías de desarrollo de software, sacando sus propios criterios de evaluación, donde se destacan metodologías ágiles.
Metodologías de desarrollo de software	[1]	Este Trabajo de Titulación tiene como objetivo realizar una comparación entre metodologías de desarrollo de software tradicionales y metodologías ágiles, y como se adaptan de manera eficiente al desarrollo de un proyecto en específico.
Estudio comparativo de metodologías tradicionales y ágiles para proyectos de Desarrollo de Software	[25]	Este estudio realiza un estudio comparativo entre metodologías tradicionales y metodologías ágiles solo de manera bibliográfica, con criterios de evaluación a nivel organizacional.

5. Metodología

Para el desarrollo del presente Trabajo de Titulación se hizo uso de varios recursos y métodos de investigación que permitieron llevar un proceso ordenado, conocer a fondo la problemática y adquirir los conocimientos necesarios para plantear una solución al problema de estudio.

Los procesos que permitió alcanzar el objetivo general del TT, se detalla en la presente sección; mostrando el contexto, proceso, los recursos y los participantes que conforman parte del TT.

5.1. Contexto

El presente Trabajo de Titulación (TT) se desarrolló en el cantón Loja, en la Universidad Nacional de Loja, en la Facultad de Energía, las industrias, y los Recursos Naturales No Renovables, en la Carrera de Ingeniería en Sistemas, donde se realizó un estudio comparativo de metodologías de desarrollo de software para aplicaciones móviles, el cual se pudo determinar con ayuda del estado del arte y un proceso experimental que se desarrolló con alumnos de noveno ciclo de la carrera de ingeniería en sistemas de la institución antes mencionada.

5.2. Proceso

El proceso de la investigación se detalla a continuación, mencionando cada uno de los objetivos con sus actividades:

Determinar las metodologías de desarrollo de software para aplicaciones móviles a evaluar y establecer los criterios de comparación.

- ✓ Revisión del estado del arte relacionado con metodologías de desarrollo de software para aplicaciones móviles usadas durante la pandemia del covid-19 **(ver Sección Resultados, Objetivo 1, Tarea 1).**
- ✓ Definición de métricas y lineamientos para la selección de trabajos relacionados con tema de investigación **(Anexo 1).**
- ✓ Recopilación de trabajos relacionados basándose en las métricas y lineamientos definidos **(Anexo 1).**
- ✓ Análisis de los trabajos seleccionados **(Anexo 1).**
- ✓ Determinar las metodologías de desarrollo de software a comparar **(ver sección Resultados, Objetivo 1, Tarea 2).**
- ✓ Definir los criterios de comparación basándose en el estado del arte **(ver sección Resultados, Objetivo 1, Tarea 2).**
- ✓ Diseño de la matriz comparativa de acuerdo a los criterios definidos **(ver sección Resultados, Objetivo 1, Tarea 2).**

Desarrollar un entorno de experimentación para la evaluación de las metodologías de desarrollo de software para aplicaciones móviles.

- ✓ Planificar un entorno de experimentación que sirva como escenario real para la evaluación de la metodología **(ver sección Resultados, Objetivo 2)**.
- ✓ Seleccionar los sujetos partícipes en la experimentación **(ver sección Resultados, Objetivo 2)**.
- ✓ Ejecutar el proceso experimental planificado con los sujetos **(ver sección Resultados, Objetivo 2)**.
- ✓ Documentar los procesos ejecutados en cada una de las fases de la metodología de desarrollo de software seleccionada **(ver sección Resultados, Objetivo 2, Anexos 2 y 3)**.
- ✓ Recopilar datos obtenidos durante la ejecución del proceso experimental **(ver sección Resultados, Objetivo 2)**.

Evaluar las metodologías de desarrollo de software para aplicaciones móviles de acuerdo a los criterios de evaluación establecidos.

- ✓ Realizar una valoración de la metodología de desarrollo establecida mediante la matriz de evaluación **(ver sección Resultados, Objetivo 3)**.
- ✓ Realizar un análisis estadístico con los resultados obtenidos en la evaluación **(ver sección Resultados, Objetivo 3)**.
- ✓ Comprobar la hipótesis de investigación de acuerdo al grado de satisfacción **(ver sección Resultados, Objetivo 3)**.
- ✓ Definir fortalezas y debilidades de la metodología de desarrollo de software evaluada **(ver sección Resultados, Fase 3)**.

5.3. Recursos

Para la propuesta a la pregunta de investigación se hizo uso de recursos científicos, académicos, técnicos y éticos, los cuales se presentan a continuación:

Recursos científicos

- **Búsqueda exploratoria:** esta técnica permitió obtener información sobre metodologías de desarrollo para aplicaciones móviles con sus respectivos ciclos de vida y sobre una guía comparativa de metodologías de desarrollo de software, que están relacionados directamente con el presente Trabajo de Titulación [2].
- **Estudio del estado del arte:** esta técnica ayudó a buscar información bibliográfica mediante una Revisión Sistemática de Literatura referente a metodologías de desarrollo móviles usadas durante la pandemia del Covid-19 **(ver Anexo 1)**, además

permitió establecer los criterios de evaluación para la construcción de la matriz comparativa. **(ver sección Resultados, Objetivo 1)**

- **Estudios de caso:** Esta técnica permitió identificar casos donde se ha realizado comparativas de metodologías de desarrollo de software en diferentes entornos de trabajo. **(ver sección Trabajos Relacionados)**
- **Método Científico:** este método fue la base principal para el desarrollo del presente Trabajo de Titulación, desde que se planteó la Propuesta de Trabajo de Titulación, hasta la culminación del mismo. Esto se puede evidenciar a través del desarrollo de las actividades planteadas de cada objetivo **(ver Sección Materiales y Métodos, apartado 5.2)**; y los resultados obtenidos en cada una de sus fases **(ver sección Resultados)**, en conjunto con la respectiva discusión **(ver sección Discusión)** y sus conclusiones **(ver sección Conclusiones)** [26].

Recursos técnicos

- **Metodología de Bárbara Kitchenham:** Con ayuda de esta metodología, fue posible realizar la Revisión Sistemática de Literatura, la cual permitió determinar las metodologías de desarrollo de software para aplicaciones móviles utilizadas durante la pandemia del Covid-19, y con esto definir los objetos experimentales a comparar (metodologías de desarrollo de software). Para ejecutar este proceso se utilizó la herramienta Parsifal, la cual, por medio de su interfaz intuitiva, facilita estas revisiones respetando el método propuesto por Bárbara Kitchenham **(Anexo 1)**.
- **Proceso Experimental:** Se siguió un modelo experimental de [26] que consta de 5 fases: Definición del alcance, Planificación, Operación, Análisis e interpretación y Presentación de resultados. Esto sirvió para desarrollar un entorno de experimentación en los objetivos 2 y 3 **(Ver sección Resultados, Objetivos 2 y 3)** donde se plantea la hipótesis relacionada con la pregunta de investigación y así determinar cuál metodología es la más adecuada para el desarrollo de aplicaciones móviles. Estas 5 fases fueron adaptadas a los objetivos del presente Trabajo de Titulación debido a que el proceso experimental fue en un campo específico como es el desarrollo de aplicaciones móviles.
- **Metodología de desarrollo por prototipos o Prototipado:** Esta metodología se aplicó con ayuda de los sujetos seleccionados en el entorno de experimentación **(ver Resultados, Objetivo 2)**, donde se desarrolló una aplicación móvil de prueba.

Recursos éticos

Consentimiento informado

Este recurso fue de gran importancia para realizar el proceso experimental en el objetivo 2, por ello se pudo obtener el consentimiento de los alumnos para participar voluntariamente en la experimentación y se dio a conocer que toda la información generada en este proceso será usada netamente para fines académicos.

Participantes

El presente trabajo de titulación fue desarrollado por los siguientes participantes:

- Andy Aron Camacho Herrera, como investigador del TT, iniciando sus actividades desde el planteamiento del problema, hasta el cumplimiento total de los objetivos planteados en este TT.
- Ing. Edison Leonardo Coronel Romero, como director del TT, orientando de manera académica y técnica en el desarrollo de los objetivos planteados para el presente TT.
- Estudiantes de noveno ciclo de la Carrera de Ingeniería en Sistemas como participantes del escenario de experimentación (**véase Resultados, Objetivo 2**).

6. Resultados

En este apartado se muestran las fases, actividades y tareas que se llevaron a cabo para desarrollar el presente Trabajo de Titulación.

En la primera fase se determina la metodología de desarrollo de software para aplicaciones móviles, en la segunda fase se desarrolla un escenario de experimentación, y en la tercera fase se realiza una evaluación sobre la metodología de desarrollo de software seleccionada, donde se determinan los puntos fuertes y débiles de la misma.

6.1. Objetivo 1: Determinar las metodologías de desarrollo de software para aplicaciones móviles a evaluar y establecer los criterios de comparación.

En esta fase se examinó el estado del arte por medio de una Revisión Sistemática de Literatura (RSL) sobre metodologías de desarrollo de software para aplicaciones móviles usadas durante la pandemia del COVID-19, con base en la metodología propuesta por Bárbara Kitchenham [27], con el fin de determinar cuáles son las metodologías de desarrollo móvil más usadas, los criterios de elección de las mismas y el sistema operativo para las que fueron desarrolladas este tipo de aplicaciones. Para realizar esta revisión se utilizó “Parsifal”, la cual es una herramienta en línea diseñada para simplificar el proceso que conlleva la RSL¹. Luego se procedió a determinar los criterios de evaluación mediante información obtenida en la RSL y los trabajos relacionados, los cuales sirvieron para construir la matriz de evaluación. La documentación como las herramientas tecnológicas se alinearon correctamente con las actividades planteadas dentro de esta fase, las mismas que se detallan a continuación.

Tarea 1: Revisar el estado del arte relacionado con metodologías de desarrollo de software para aplicaciones móviles usadas durante la pandemia del Covid-19.

En esta actividad se realizó una Revisión Sistemática de Literatura (**véase Anexo 1**), donde su objetivo principal es identificar las metodologías de desarrollo de software para aplicaciones móviles más utilizadas durante la pandemia de covid-19, es importante mencionar que las búsquedas solo se limitaron desde al año 2020 hasta mayo 2021, debido a que ese tiempo fue el más crítico de la pandemia. Además, se obtuvo los criterios de selección de la metodología que los respectivos autores mencionan, y el sistema operativo en el cual fue desarrollada la aplicación.

El proceso desarrollado en esta RSL fue sintetizado en 7 pasos, como se puede observar en la Figura 7, donde se describe cada una de estas fases que se ejecutó en este proceso aplicando el método propuesto por Bárbara Kitchenham [27].

¹ <https://parsifal.ai>

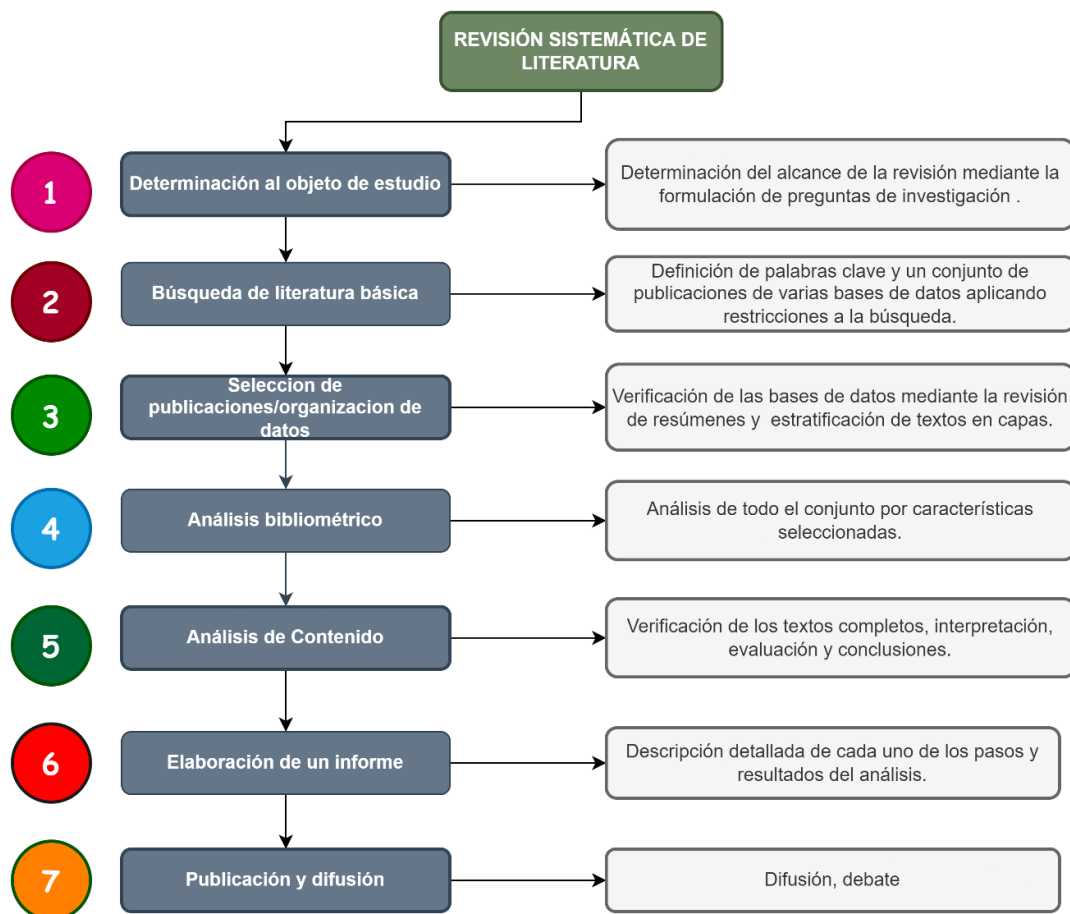


Figura 7. Proceso de la Revisión Sistemática de Literatura

Preguntas de investigación

Para obtener datos que nos ayuden a cumplir con los objetivos del presente TT, en esta RSL, se establecieron 3 preguntas de investigación (véase Tabla 3), las cuales fueron de gran importancia en todo el proceso ejecutado en este trabajo.

Tabla 3.
Preguntas de Investigación de la RSL

ID	Preguntas de Investigación
P1	¿Qué metodologías de desarrollo de software se usaron durante la pandemia del covid-19 para crear aplicaciones móviles?
P2	¿Qué criterios mencionan los autores para la elección de una metodología de desarrollo para aplicaciones móviles?
P3	¿Para qué tipo de Sistema Operativo fueron desarrolladas las aplicaciones móviles de acuerdo a los estudios seleccionados?

Palabras claves

Para la realización de la RSL se utilizó la siguiente lista de palabras claves, ya que estas permitieron a realizar una búsqueda de información más exhaustiva y eficiente:

- Software developement process
- University
- Companies
- Development methodology
- Agile methodology
- Mobile devices
- Mobile application
- Coronavirus

Cadenas de Búsqueda

Para las cadenas de búsqueda estructuradas en esta RSL, se utilizó la semántica SQL, aplicando operadores Lógicos AND, OR y las cadenas de búsqueda. Con todo esto se procedió a construir tres cadenas como se puede observar en la Tabla 4 . Estas cadenas se la utilizaron en cada una de las bases de datos científicas seleccionadas.

Tabla 4.
Cadenas de búsqueda

ID	Cadenas de Búsqueda
CB_1	(university OR institute OR companies) AND ("agile methodologies" OR methods) AND ("mobile devices" OR "mobile applications")
CB_2	(university OR institute OR companies) AND ("agile methodologies" OR methods) AND ("mobile devices" OR "mobile applications") AND (Coronavirus)
CB_3	("software development process" OR "agile methodology" OR "development methodology") AND ("Software Development") AND ("mobile" OR "mobile application")

Bases de Datos

Para las búsquedas se utilizó 4 Bases de Datos Científicas como se puede observar en la Tabla 5.

Tabla 5.
Bases de datos científicas

Base de Datos	URL
ISI Web of Science	http://www.isiknowledge.com
Scopus	http://scopus.com/
IEEE Digital Library	https://ieeexplore.ieee.org/
ACM Digital Library	https://dl.acm.org/

Criterios de Selección

Criterios de Inclusión: Los criterios de inclusión aplicados se los puede observar en la Tabla 6, respectivamente con su descripción.

Tabla 6.
Criterios de inclusión

Criterio	Descripción
Contenido	Debe contener al menos en una de las cadenas de búsqueda en su título y una o algunas en su resumen.
Fecha de publicación	Estudios publicados desde el año 2020-2021
Motores de búsqueda	WoS, Scopus, IEEE Xplore, ACM.
Idioma	Se consideran artículos estrictamente en inglés, tomando en cuenta un porcentaje bajo en español.
Tipos de estudios	Artículos de revistas y artículos de conferencias.

Criterios de exclusión: No consideraron los estudios que no cuenten con información clave que aporten a responder las preguntas de investigación y que no cumplan con los criterios de inclusión, que se muestran en la Tabla 6.

Ejecución de las cadenas de búsqueda

En la Tabla 7, se puede observar las cadenas de búsqueda ejecutadas en cada una de las bases de datos científicas seleccionadas para esta RSL, las cuales dieron un resultado de un total de 308 estudios.

Tabla 7.
Ejecución de las cadenas de búsqueda

Base de Datos	Cadena de Búsqueda	Número de Resultados
SCOPUS	("university" OR "companies") AND ("software development process" OR "agile methodology" OR "development methodologies") AND ("Software Development") AND ("mobile" OR "mobile application") AND (LIMITTO (OA , "all")) AND (LIMITTO (PUBYEAR , 2021) OR LIMITTO (PUBYEAR , 2020))	154
IEEE Xplorer	((("All Metadata":university OR companies) AND "All Metadata":mobile o mobile aplicación) AND "All Metadata":"software development process" OR "agile methodology" OR "development methodology")	50
WoS	ALL=(university OR companies) AND ALL=("software development process" OR "agile methodology" OR "development methodology") AND ALL=("mobile" OR "mobile aplicación")	15
ACM	AllField:(university OR companies) AND AllField:("software development process" OR "agile methodology" OR "development methodology") AND AllField:(mobile OR "mobile aplicación")	89
	TOTAL	308

Proceso de selección de estudios primarios

Para realizar la selección de los estudios primarios se realizó un proceso el cual se puede evidenciar en la Figura 8.

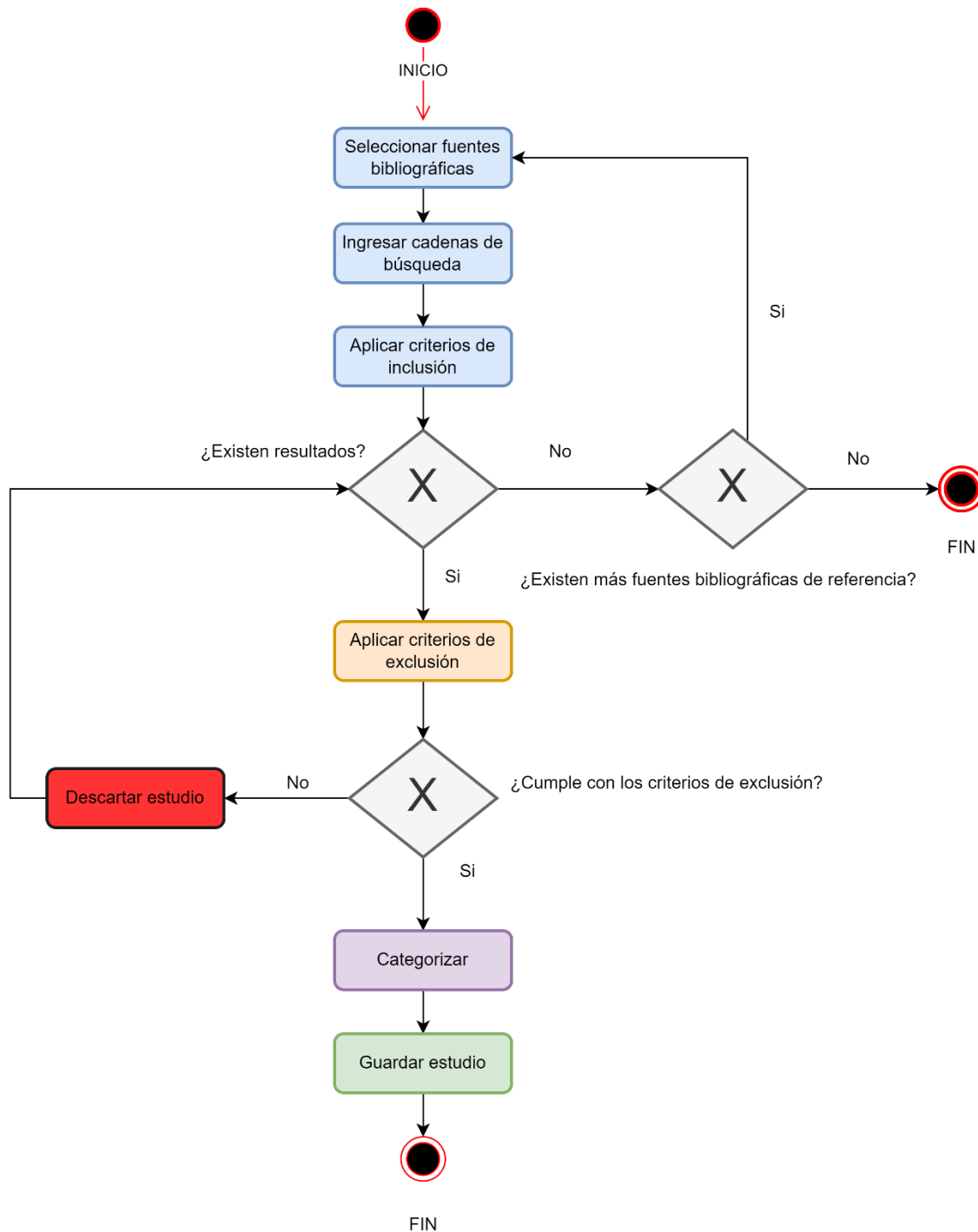


Figura 8. Diagrama de Flujo del Proceso de selección de artículos primarios

Síntesis de datos

Para extraer la información que nos ayude a responder las 3 preguntas planteadas en esta RSL (ver Tabla 3), se procedió a realizar los análisis respectivos. Este proceso de análisis ayudó a seleccionar los estudios finales, los cuales serían objetos de estudio y por ende en la extracción de datos. Los análisis ejecutados en este proceso se los puede observar en la Figura 9.

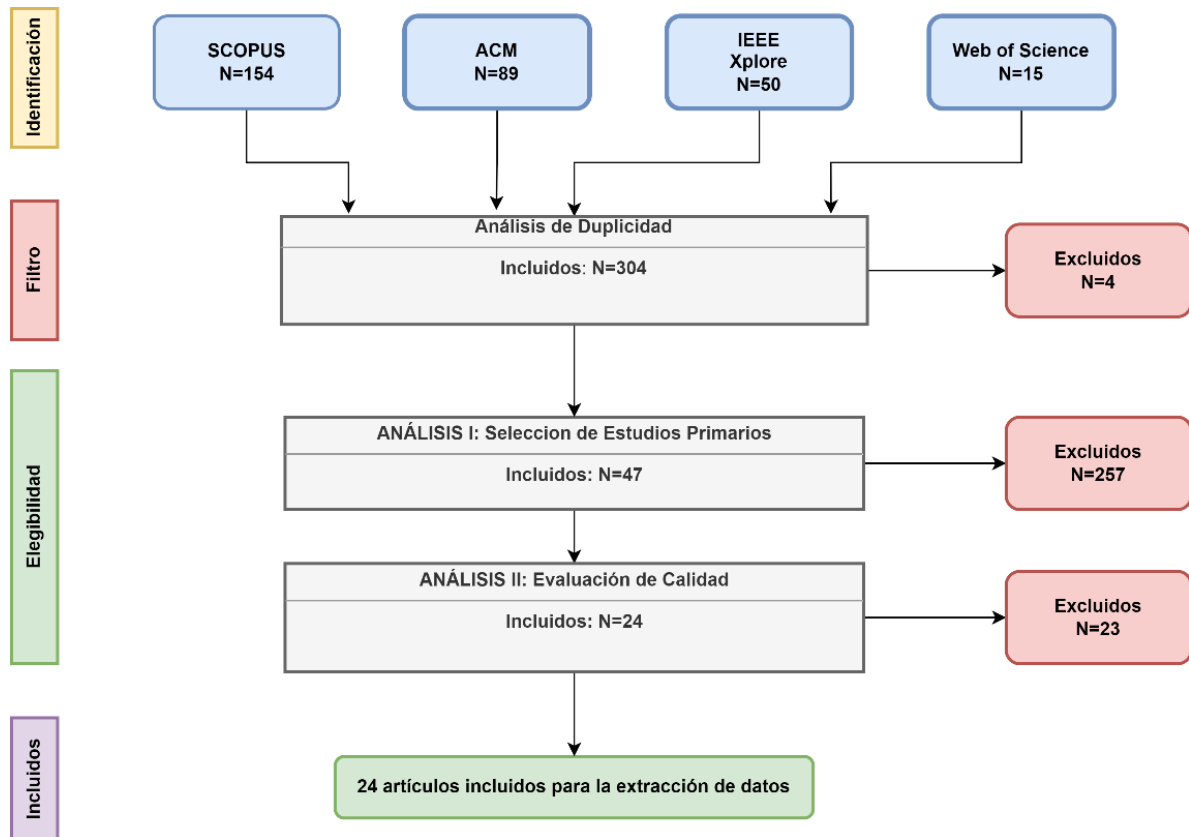


Figura 9. Síntesis del proceso de análisis para los estudios finales

Como resultado de los análisis mencionados anteriormente, se obtuvo un total de 24 estudios finales. En la Tabla 8, se muestra detalladamente los estudios seleccionados, respectivamente con la información extraída que responda las preguntas de investigación.

Tabla 8.
Estudios seleccionados

Nombre del estudio	Base de Datos	Metodología aplicada	Criterios de selección	Sistema operativo
Automated Test Input Generation via Model Inference Based on User Story and Acceptance Criteria for Mobile Application Development	Web of Science	Scrum	calidad del producto, ciclos cortos, desarrollo ágil	Multiplataforma
An Agile-Based Integrated Framework for Mobile Application Development Considering Illities	Web of Science	Scrum	capacidad de agilidad, satisfacción del cliente	Android
EmERGE mHealth Platform: Implementation and Technical Evaluation of a Digital Supported Pathway of Care for Medically Stable HIV	Web of Science	Hybrid(Scrum-XP)	desarrollo incremental, calidad del producto, adaptable a cambios, proyectos pequeños/medios	Multiplataforma
Realtime Notifications on Visitor Tracking Systems Using Android and Arduino	SCOPUS	Cascada	calidad del producto	Android
Thailand medical mobile application for patients triage base on criteria based dispatch protocol	SCOPUS	Cascada	satisfacción del cliente	Multiplataforma
CardioResyncApp: A mobile phone application for recording research data in Cardiology [CardioResyncApp: Un aplicativo móvil para recolectar datos de investigación en Cardiología]	SCOPUS	Scrum	ciclos cortos, desarrollo incremental, colaboración, capacidad de agilidad	Multiplataforma
The Necessity of Interdisciplinary Software Development for Building Viable Research Platforms: Case Study in Automated Drug Delivery in Diabetes	ACM	Prototipado	desarrollo iterativo, calidad del producto, proyectos grandes	Android
Developing a Design Phase for a Mentoring Mobile App	ACM	Prototipado	desarrollo iterativo	Multiplataforma
Close2U: An App for Monitoring Cancer Patients with Enriched Information from Interaction Patterns	Web of Science	Prototipado	desarrollo iterativo e incremental	Android

Using goal-directed design to create a mobile health app to improve patient compliance with hypertension self-management: Development and deployment	SCOPUS	Prototipado	desarrollo incremental, producto funcional	Multiplataforma
Systems dynamics and serious video games in an appropriation strategy of a decision support system for small livestock farmers	SCOPUS	RUP	documentación adecuada, calidad del producto, desarrollo iterativo, proyectos grandes	Android
Multiplatform system to improve tourist services in the city of trujillo [Sistema Multiplataforma para mejorar los Servicios Turísticos en la ciudad de Trujillo]	SCOPUS	Hybrid(Scrum-XP)	colaboración, capacidad de agilidad	Multiplataforma
Mobile application to improve tourist flow in trujillo [Aplicación Móvil para Mejorar el Flujo Turístico en Trujillo]	SCOPUS	Mobile-D	capacidad de agilidad, calidad del producto	Android
Investigating agile adoption in saudi arabian mobile application development	SCOPUS	Mobile-D	proyectos pequeños/medios, desarrollo ágil	Multiplataforma
Doodle2App: Native app code by freehand UI sketching	SCOPUS	Prototipado	proyectos grandes	Android
Application Information System Smart Parking Based on Smartphone	SCOPUS	Cascada	calidad del producto	iOS
A Mobile Serious Game About the Pandemic (COVID-19 - Did You Know?): Design and Evaluation Study	SCOPUS	Scrum	desarrollo iterativo, desarrollo incremental	Multiplataforma
A software to prevent delirium in hospitalised older adults: Development and feasibility assessment	SCOPUS	Scrum	colaboración, roles intercambiables, adaptable a cambios	Android
A mobile health approach for improving outcomes in suicide prevention (SafePlan)	SCOPUS	Prototipado	proyectos grandes	Multiplataforma
A comparative analysis of RAD and agile technique for management of computing graduation projects	SCOPUS	RAD	documentación adecuada	Android

Design of a mobile application for the school enrollment process in order to prevent Covid-19	SCOPUS	Scrum	adaptable a cambios, satisfacción del cliente	Android
Troubadour: A Gamified e-Learning Platform for Ear Training	Web of Science	Prototipado	satisfacción del cliente	Android
Agile Methodology Advantages when delivering Internet of Things projects	IEEE	Scrum	desarrollo incremental, desarrollo iterativo, colaboración en equipo, roles intercambiables	Multiplataforma
Scrum to Support Application Development Project for Online Learning	ACM	Scrum	adaptable a cambios, satisfacción del cliente	Android

Análisis de los resultados

Para los resultados, se procedió a realizar el análisis de los mismos de acuerdo a cada pregunta planteada en la RSL, esto se detalla a continuación:

Pregunta 1: ¿Qué metodologías de desarrollo de software se usaron durante la pandemia del covid-19 para crear aplicaciones móviles?

De acuerdo a los 24 estudios seleccionados, se tiene que las metodologías de desarrollo software más utilizados durante la pandemia para la creación de aplicaciones móviles son: Scrum, Prototipado, Cascada, RUP, RAD, Mobile-D y una metodología híbrida que combina Scrum y XP.

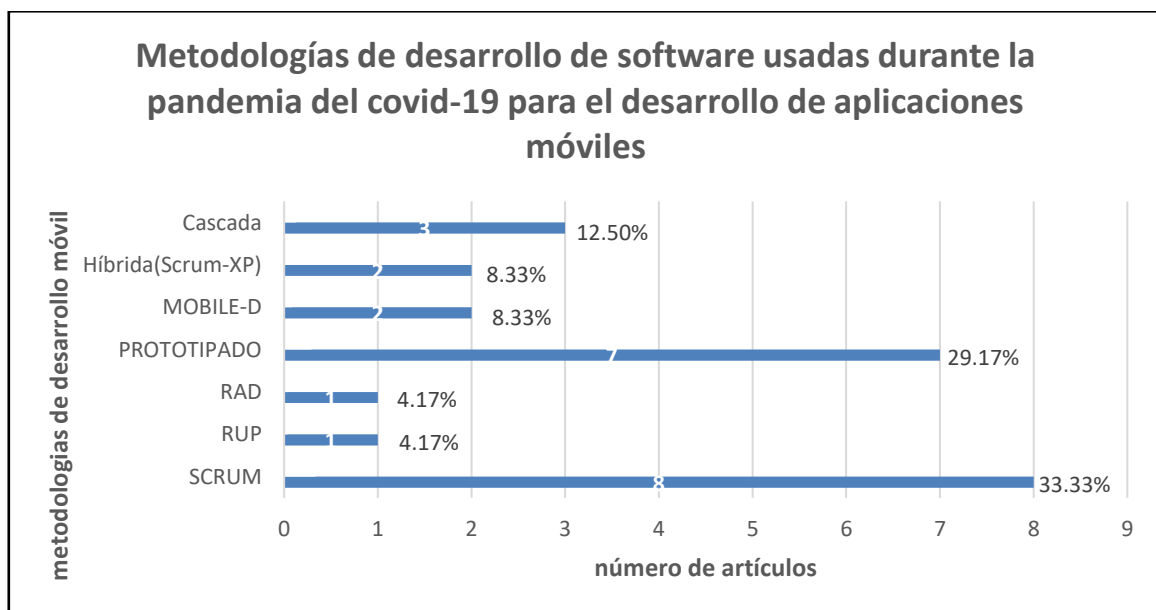


Figura 10. Metodologías de desarrollo de software móvil usadas durante la pandemia del covid-19.

Como se puede evidenciar en la Figura 10, el framework Scrum ha sido el marco de trabajo más utilizado para la gestión de proyectos referentes al desarrollo de aplicaciones móviles con un equivalente del 33.33% de los estudios seleccionados y la metodología Prototipado con un 29.17%.

Pregunta 2: ¿Qué criterios mencionan los autores para la elección de una metodología de desarrollo para aplicaciones móviles?

Para responder a esta pregunta de investigación; se han analizado completamente los 24 estudios seleccionados y se ha recopilado los criterios de selección de la metodología que cada autor menciona en su artículo. Como resultado se obtuvo 12 criterios, los cuales son: roles intercambiables, documentación adecuada, producto funcional, proyectos grandes, colaboración en equipo, proyectos pequeños/medianos, adaptable a cambios, desarrollo iterativo e incremental, satisfacción del cliente, desarrollo ágil, ciclos cortos, calidad del producto.

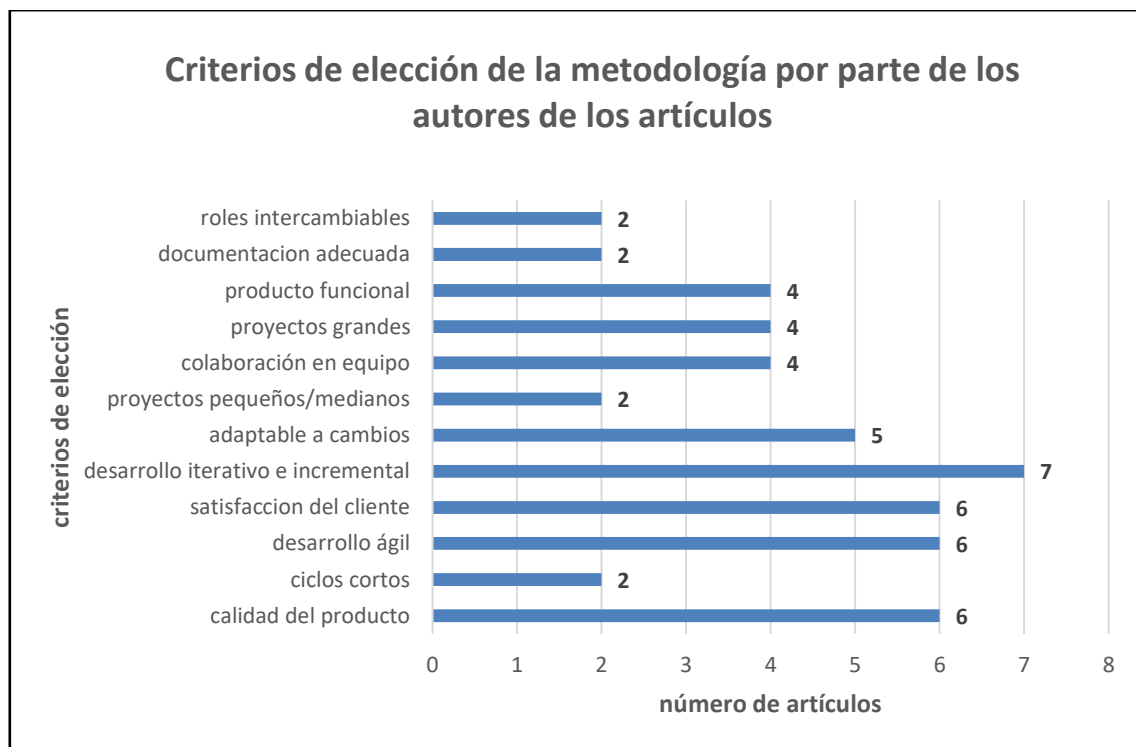


Figura 11. Criterios de elección de las metodologías de desarrollo de software móvil.

Como se puede observar en la Figura 11, se muestran los criterios que varios autores proponen al momento de elegir una metodología de desarrollo móvil. Los criterios que más consideran son: desarrollo iterativo e incremental, satisfacción al cliente, desarrollo ágil y calidad del producto.

Pregunta 3: ¿Para qué tipo de Sistema Operativo fueron desarrolladas las aplicaciones móviles de acuerdo a los estudios seleccionados?

En la Figura 12 se muestran los sistemas operativos en los que fueron desarrollados las aplicaciones móviles utilizando las metodologías de desarrollo de software mostradas anteriormente. Se puede observar que las aplicaciones fueron desarrolladas para sistemas operativos Android, iOS y Multiplataforma.

Sistemas Operativos para el desarrollo de aplicaciones móviles durante la pandemia del covid-19

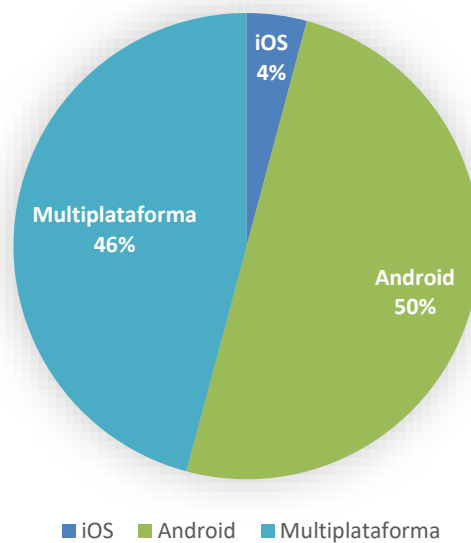


Figura 12. Sistemas operativos para el desarrollo móvil

Como se puede observar en la Figura 12, Android lidera con un 50% de los estudios seleccionados, es decir, la mitad de los autores consideraron desarrollar sus aplicaciones en Android. Cabe recalcar que el 46% consideraron desarrollar aplicaciones móviles multiplataforma. Estos resultados son de gran importancia debido a que se los tomó en cuenta para el desarrollo de una aplicación móvil que sirvió como escenario de experimentación (**véase Objetivo 2**).

Tarea 2: Determinar la metodología de desarrollo de software a evaluar y los criterios de valoración.

Para determinar la metodología a evaluar, se tomó en cuenta los resultados de la Revisión Sistemática de Literatura (**véase Objetivo 1**) donde Prototipado es la metodología de desarrollo de software más usada. Es importante mencionar que Scrum fue descartado debido

a que, según la documentación oficial de Scrum², se trata de un framework para la gestión de proyectos y no una metodología de desarrollo de software.

Para el análisis de evaluación se ha construido una matriz siguiendo una guía comparativa de [2], de igual manera toda la información para los criterios de comparación se ha tomado de los trabajos [5] [22] [28] [29]; y se complementó y sustentó con la RSL.

La matriz se ha construido complementando varios criterios, los cuales se clasificaron en diferentes categorías:

Usabilidad: Los criterios planteados en la Tabla 9, tratan de evaluar todos los beneficios que el equipo de desarrollo y el cliente obtiene utilizando estas metodologías como: incremento de la productividad, calidad y satisfacción.

Tabla 9.
Criterios por usabilidad

Usabilidad	Descripción
Adaptación a entornos turbulentos	La metodología se adapta a una forma de trabajo poco común y con riesgos que pueden afectar el proyecto.
Satisfacción del cliente	El cliente se encuentra satisfecho con el producto final.
Estándares de calidad	La metodología aplicada se basa en estándares que otorguen más calidad al producto.
Respeto de las fechas de entregas	Se cumple con las fechas de entrega del producto.
Cumplimiento de requerimientos	Se cumple con todos los requerimientos establecidos por el cliente, haciéndose mediante pruebas.
Trabajo remoto	La metodología se adapta a la modalidad de trabajo remoto.
Herramientas	Existen múltiples herramientas que faciliten los procesos que conlleva la metodología.

² Scrum.org

Aplicabilidad: Estos criterios representan cuando un entorno es favorable para la aplicación de las metodologías. Los criterios se muestran en la Tabla 10.

Tabla 10.
Aplicabilidad

Aplicabilidad	Descripción
Interacción entre los miembros el equipo	La metodología permite que los miembros del equipo interactúen entre sí, con el fin de ejecutar el proyecto de la mejor manera.
Interacción con el cliente	La metodología permite que algunos o todos los miembros del equipo interactúen con el cliente, esto con el objetivo de entregar un producto de calidad y cumpliendo con todos los requerimientos al cliente.
Evaluación de riesgos del proyecto	Los miembros del equipo realizan evaluaciones sobre los riesgos que pueden surgir durante la ejecución del proyecto.
Aplicable para proyectos pequeños	Es posible aplicar la metodología seleccionada en proyectos que no requieran muchos recursos, ni tiempo.
Requiere una buena organización del equipo.	La metodología permite establecer roles y organizar las tareas para cada uno de estos.
Aplicable en equipos pequeños	Se puede aplicar la metodología con un equipo pequeño de 5 personas.
Adaptable a cambios	La metodología se adapta sin ningún problema a cambios requeridos por el cliente.
Basados en casos de uso	Los miembros del equipo utilizan casos de uso para describir de una mejor manera los requerimientos.
Documentación adecuada	Los miembros del equipo documentan todos los procesos ejecutados durante el desarrollo del proyecto.

Integración entre las etapas de desarrollo.	La metodología permite conectar o unificar sus fases para facilitar los procesos.
Relación con UML	Los miembros del equipo utilizan UML como lenguaje para modelar la arquitectura, diseño e implementación del sistema.
Permite el desarrollo de software para aplicaciones móviles	La metodología permite adaptarse a cualquier tipo de proyecto de desarrollo de software en aplicaciones móviles.
Aplicable en tiempos pandémicos	Se puede aplicar la metodología bajo efectos y restricciones de la pandemia del covid-19.

Procesos y productos: Representan cómo se caracterizan las metodologías como tal, donde se definen las actividades de desarrollo, las pruebas y los productos que se obtienen. Estos criterios se muestran en la Tabla 11.

Tabla 11.
Procesos y productos

Procesos y Productos	Descripción
Pruebas unitarias o de escenarios	El equipo de desarrollo realiza pruebas unitarias o de escenarios.
Pruebas de aceptación	El equipo de desarrollo realiza pruebas de aceptación en conjunto con el cliente.
Control de Calidad	El equipo de desarrollo realiza un seguimiento en la calidad del producto.
Producto funcional /Ejecutable	El equipo de desarrollo entrega un producto funcional al cliente.

Capacidad: En la Tabla 12, se muestran los criterios sobre la capacidad de las metodologías respecto a cambios durante el desarrollo y al ambiente de trabajo.

**Tabla 12.
Capacidad**

Capacidad	Descripción
Los requisitos funcionales pueden cambiar	La metodología permite cambiar los requisitos funcionales obtenidos durante el análisis de requerimientos.
Requisitos no funcionales pueden cambiar	La metodología permite cambiar los requisitos no funcionales.
Roles intercambiables	La metodología aplicada es flexible en la organización del equipo, permitiendo cambiar de roles y por ende tareas.
Integración de los cambios	La metodología permite realizar cambios sin que afecte a los procesos de las demás fases de la metodología aplicada.
Conocimiento compartido	Existe un conocimiento compartido entre los miembros del equipo, lo cual disminuye la incertidumbre en la ejecución del proyecto.
Colaboración	La metodología permite que exista una colaboración en la ejecución de las tareas que tiene cada miembro del equipo.
Ambiente de trabajo favorable	La metodología asegura de que exista un buen ambiente de trabajo entre los miembros del equipo y el cliente.
Posibilidad de reactividad (cada etapa, cada iteración)	La metodología permite fácilmente reactivar o reprogramar las actividades en cada Sprint.
Iteraciones cortas	La metodología tiene ciclos pequeños que no llevan más de 4 semanas.
El plan de trabajo se puede cambiar	El plan de trabajo establecido se puede cambiar fácilmente.

Una vez establecidos los criterios de comparación, se procedió a construir la matriz comparativa unificando todos estos criterios como se muestra en la Tabla 13, compuesta por 4 categorías y un total de 34 criterios de comparación.

Tabla 13.
Matriz comparativa

Categorías	Métricas	Metodología de Desarrollo
		PROTOTIPADO
USABILIDAD	Adaptarse a entornos turbulentos	-
	Satisfacción del cliente	-
	Respeto de un nivel de calidad	-
	Respeto de las fechas de entregas	-
	Cumplimiento de requerimientos	-
	Trabajo remoto	-
	Herramientas	
APLICABILIDAD	Interacción entre los miembros el equipo	-
	Interacción con el cliente	-
	Evaluación de riesgos del proyecto	-
	Aplicable para proyectos pequeños	-
	Requiere una buena organización del equipo	-
	Aplicable en equipos pequeños	-
	Adaptable a cambios	-
	Basados en casos de uso	-
	Documentación adecuada	-
	Integración entre las etapas de desarrollo	-

	Relación con UML	-
	Permite el desarrollo de software para aplicaciones móviles	-
	Aplicable en tiempos pandémicos	-
PROCESOS Y PRODUCTOS	Pruebas unitarias o de escenarios	-
	Pruebas de aceptación	-
	Control de Calidad	-
	Ejecutable	-
CAPACIDAD	Los requisitos funcionales pueden cambiar	-
	Requisitos no funcionales pueden cambiar	-
	Roles intercambiables	-
	Integración de los cambios	-
	Conocimiento compartido	-
	Colaboración	-
	Ambiente de trabajo favorable	-
	Posibilidad de reactividad (cada etapa, cada iteración)	-
	Iteraciones cortas	-
	El plan de trabajo se puede cambiar	-

6.2. Objetivo 2: Desarrollar un entorno de experimentación para la evaluación de las metodologías de desarrollo de software para aplicaciones móviles.

Para realizar la experimentación se tomó como base el libro titulado “Métodos de investigación de ingeniería del software, Capítulo 3” [26], el cual explica el procedimiento que se debe seguir para realizar una experimentación válida. Además, indica todas las actividades y tareas que conlleva este proceso experimental. Estas fases fueron adaptadas para esta experimentación.

Proceso Experimental

La realización de un experimento es la manera más apropiada para responder a la pregunta de investigación, para ello se tomó en cuenta ciertos aspectos fundamentales basados en un modelo de [26], como se puede observar en la Figura 13.

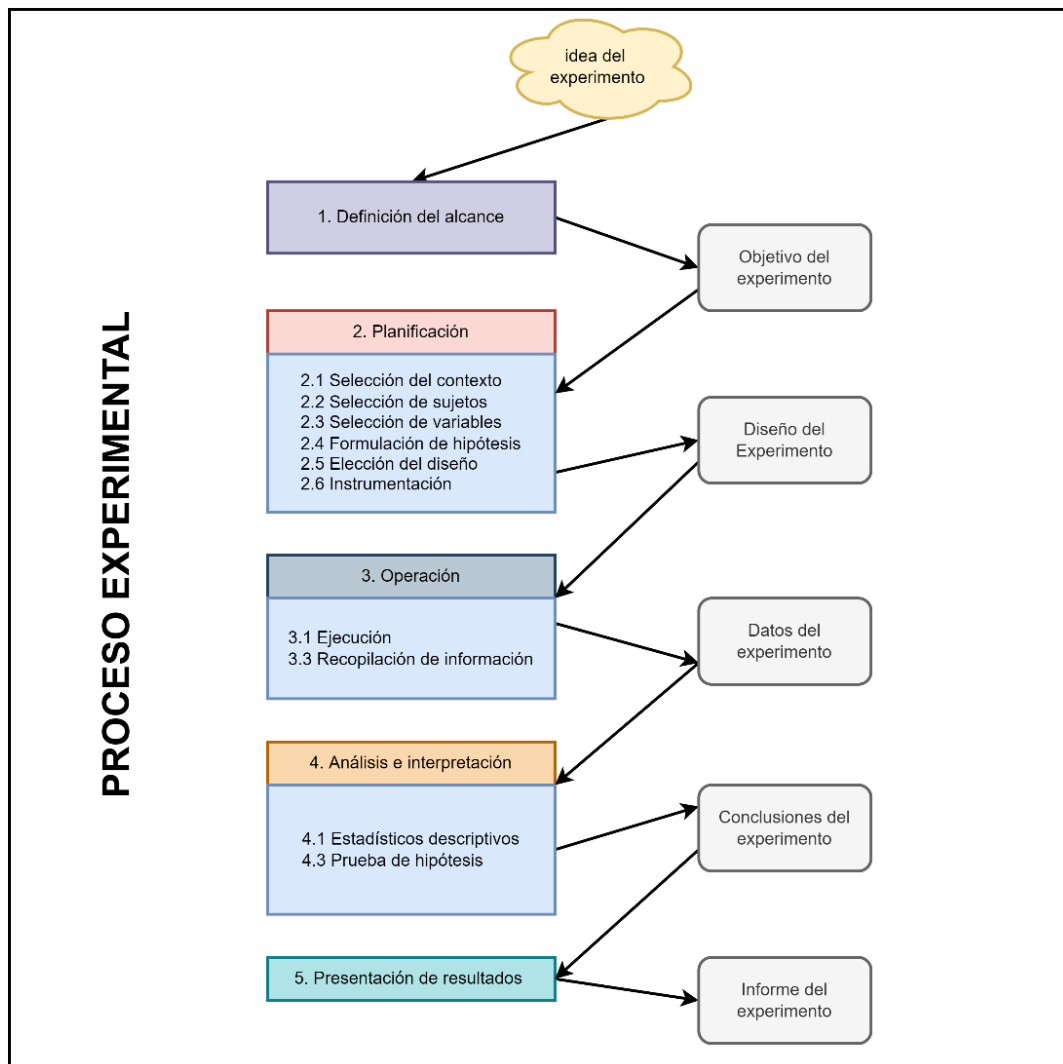


Figura 13. Proceso Experimental [26].

Definición del alcance

El objetivo del experimento es “Aplicar la metodología Prototipado mediante el desarrollo de una aplicación móvil de prueba, en el contexto de los alumnos de noveno ciclo de la carrera de Ingeniería en Sistemas, de la Universidad Nacional de Loja”.

Se definió que la evaluación se llevará a cabo por medio de una ampliación móvil para pedidos de comida, debido a que según un estudio de la Universidad Politécnica Salesiana del Ecuador sede en Guayaquil [30], este tipo de aplicaciones son las más demandadas durante la pandemia del Covid-19, además esta información se complementa con páginas web [31] [32] [33] que se dedican a estudios de rating de aplicaciones.

En el presente objetivo la experimentación se realizará hasta el punto de Operación y en el objetivo 3, se continúa con los siguientes puntos: Análisis e interpretación, y presentación de resultados.

Planificación

Selección del Contexto

El objeto experimental consiste en la metodología de desarrollo para aplicaciones móviles Prototipado. Esta metodología fue seleccionada de acuerdo a los resultados del objetivo 1 del presente TT. La metodología fue aplicada en un escenario de experimentación, donde se evaluó en un grupo de estudiantes los cuales son aptos para el proyecto. Los estudiantes desarrollaron la aplicación requerida donde tuvieron que aplicar la ingeniería de requerimientos de acuerdo a la metodología establecida, esto con el fin de analizar los factores claves que ayudan a la valoración del estudio.

Selección de Sujetos

En el experimento participaron alumnos de noveno ciclo de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja. Estos sujetos se consideraron aptos para el experimento debido a que se tomó como base su malla curricular donde reciben materias que imparten conocimiento sobre análisis de sistemas, diseño de sistemas, desarrollo de aplicaciones.

Los sujetos tuvieron un máximo de 5 semanas para cumplir con la aplicación requerida, respetando las fases de desarrollo que conlleva realizar una aplicación.

Variables

- Variable Independiente:

Metodología de desarrollo de software Prototipado

- Variable Dependiente:

Desarrollo de aplicaciones móviles

Formulación de Hipótesis

Para esta investigación se formula la siguiente hipótesis relacionadas con las variables establecidas en esta experimentación:

- **H_i**: Prototipado es una metodología de desarrollo de software que se adapta al desarrollo de aplicaciones móviles en tiempos de pandemia del covid-19.

Instrumentación

En la instrumentación se diseñó una directriz para, la cual consta de una serie de tareas a cumplir de acuerdo a la metodología correspondiente (ver Tabla 14Tabla). Estas tareas fueron diseñadas con la finalidad de ver características, ventajas, y desventajas a evaluar, así logrando comprobar la hipótesis planteada.

Tabla 14.
Resumen de tareas

ID	Descripción
T01	Realizar una aplicación móvil para realizar pedidos de comida.
T02	Realizar una aplicación móvil para dispositivos Android.
T03	Se deben aplicar todas las fases de la metodología correspondiente.
T04	Documentar todos los procesos realizados durante la creación de la aplicación.
T05	Cumplir con el tiempo de entrega (tiempo max. 5 semanas).
T06	Subir el código fuente a un repositorio.
T07	Reunirse con el cliente por la menos 1 vez por semana.

Operación

Se ha creado un escenario de experimentación donde se muestra las fases de la metodología aplicada, así mismo los materiales y herramientas utilizadas para el desarrollo de la aplicación móvil.




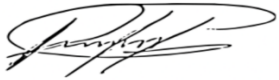
Escenario de experimentación con Prototipado

Roles

Para definir los roles en este tipo de metodología, se estableció una reunión virtual por la plataforma zoom, donde se determinó los roles de acuerdo a sus habilidades y experiencias al trabajar con esta metodología.

Los roles de cada integrante, así como su firma de aceptación para participar en el proyecto, se muestran en la Tabla 15.

Tabla 15.
Roles de Prototipado

PROTOTYPING TEAM		
NOMBRE Y APELLIDOS	ROL	Firma de Aceptación
Pavel Rivadeneira	Analista	 Pavel Alexander Rivadeneira Rojas C.I. 1724002009
Byron Stalin Montaña Beltrán	Diseñador	
Jorge Luis Pucha Vanegas	Programador	
Angel Ricardo Capa Medina	Tester	

Especificación de requerimientos

La especificación de requerimientos permitió identificar las funcionalidades requeridas por el cliente para la aplicación de prueba.

Requerimientos funcionales y no funcionales

En la Tabla 16, se muestran los requerimientos funcionales y no funcionales de la aplicación móvil.

Tabla 16.
Requerimientos de la aplicación de prueba

REQUERIMIENTOS FUNCIONALES	
RF_1	El usuario podrá registrarse en la aplicación
RF_2	El usuario tendrá que autenticarse como cliente o restaurante para hacer uso de la app.
RF_3	La aplicación permitirá al administrador crear, modificar, y listar los datos del usuario.
RF_4	La aplicación permitirá al administrador crear, modificar, eliminar menús de los restaurantes.
RF_5	La aplicación permitirá al administrador ingresar, modificar, listar, eliminar productos de los restaurantes.
RF_6	El cliente podrá visualizar el listado de restaurantes y sus productos.
RF_7	El cliente podrá agregar, modificar, o eliminar productos del carrito de compras.
RF_8	El cliente podrá dar una calificación a los restaurantes.
RF_9	La aplicación mostrará el valor total a pagar de los productos añadidos al carrito de compras.
RF_10	El cliente tendrá la opción de elegir si retira su pedido o se le envía a domicilio.
REQUERIMIENTOS NO FUNCIONALES	
Seguridad	La aplicación tendrá autenticación de usuarios
Usabilidad	La aplicación tendrá interfaces intuitivas y fáciles de comprender
Rendimiento	La aplicación tendrá tiempos de respuestas óptimos que puedan satisfacer las expectativas del usuario.

Casos de uso del Prototipo versión 1.0

El analista y diseñador optaron por modelar los requerimientos por medio de casos de uso, con el fin de modelar la funcionalidad del sistema. Los casos de uso sirven para identificar los agentes externos denominados actores y cómo estos interactúan con el sistema [34].

Para la primera versión del prototipo se diseñaron 3 diagramas de caso de uso como se puede observar en la Figura 14, Figura 15, Figura 16.

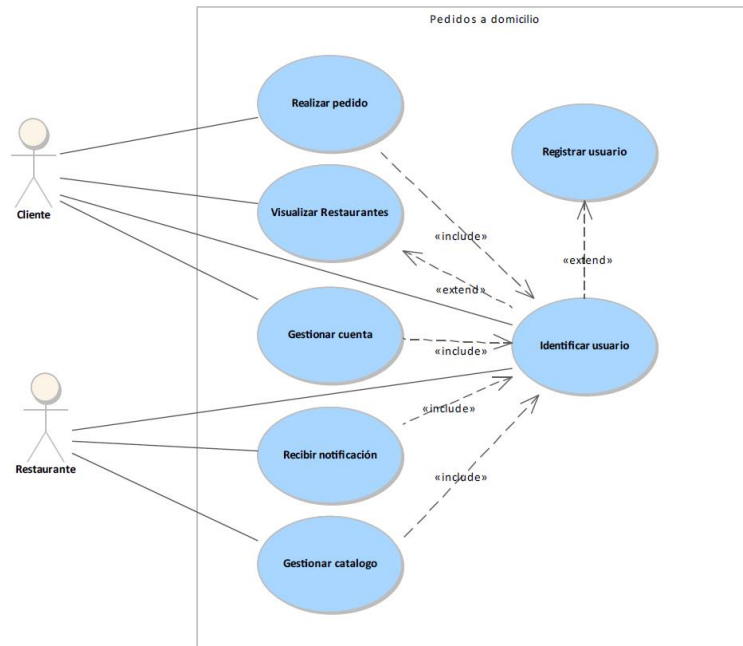


Figura 14. Diagramas de Casos de Uso general

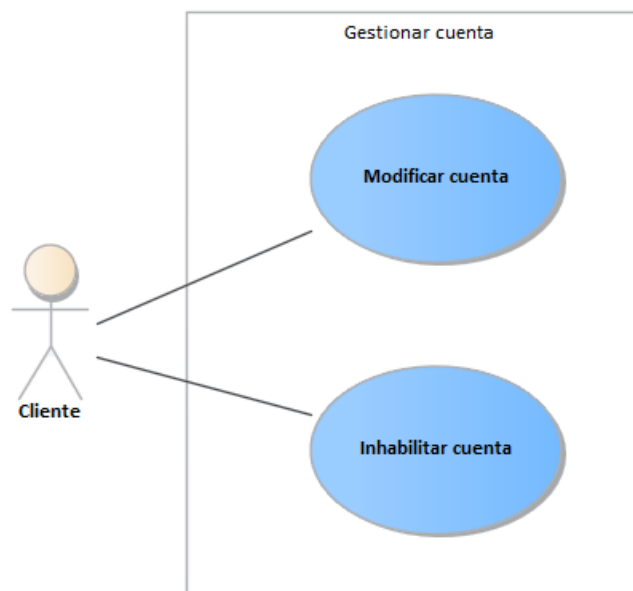


Figura 15. Diagrama de Caso de Uso-Gestionar Cuenta

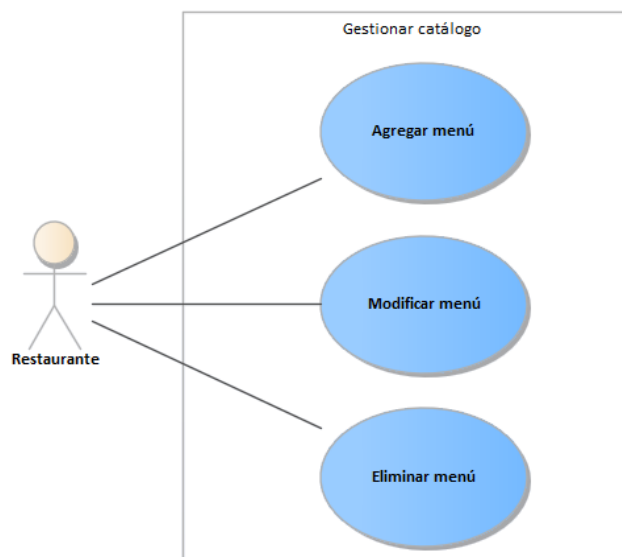


Figura 16. Diagrama de Caso de Uso-Gestionar Catálogo

Descripción de los casos de uso (Prototipo v1.0)

Tabla 17.
CU01-Registrar Usuario

Nombre del caso de uso: Registrar usuario		Id Único: CU-01
Versión: 1.0		
Actor(es): Cliente, Restaurante, Administrador		
Descripción: Permitir que los usuarios clientes y usuarios restaurantes se registren para que puedan tener su cuenta.		
Evento desencadenador: El usuario ingresa a la ventana de registro, introduce sus nombres, correo electrónico y contraseña; y hace clic en el botón registrarse.		
Pasos realizados (ruta principal)		Información para los pasos
1. El usuario ingresa a la aplicación.		Aplicación
2. El usuario introduce su información en el formulario de registro y hace clic en el botón registrarse .		Formulario de registro, nombres, apellidos, dirección, teléfono, correo electrónico, contraseña.
3. Se valida la información del usuario		Formulario de registro
Precondiciones: El usuario ingresó a la aplicación.		
Requerimientos cumplidos: Permitir que los usuarios puedan registrarse en la aplicación.		
Excepciones:		
3.1. El usuario ingresa información incorrecta		

Tabla 18.
CU02-Identificar Usuario

Nombre del caso de uso: Identificar usuario	Id Único: CU-02
Versión: 1.0	
Actor(es): Cliente, Restaurante, Administrador	
Descripción: Permitir que los usuarios clientes y usuarios restaurantes puedan iniciar sesión en la aplicación.	
Evento desencadenador: El usuario ingresa su correo electrónico y contraseña; y hace clic en el botón iniciar sesión.	
Pasos realizados (ruta principal)	Información para los pasos
1. El usuario ingresa a la aplicación.	Aplicación
2. El usuario introduce su información en el formulario de registro y hace clic en el botón registrarse .	Formulario de inicio de sesión, correo, contraseña.
3. Se valida la información del usuario	Formulario de inicio de sesión
Precondiciones: El usuario se ha registrado en la aplicación.	
Requerimientos cumplidos: Permitir que los usuarios puedan iniciar sesión en la aplicación.	
Excepciones:	
3.1. El usuario ingresa información incorrecta	
3.2. El usuario no se encuentra registrado	

Tabla 19.
CU03-Realizar Pedido

Nombre del caso de uso: Realizar pedido	Id Único: CU-03
Versión: 1.0	
Actor(es): Cliente	
Descripción: Permitir que los clientes puedan realizar pedidos a los restaurantes disponibles en la aplicación.	
Evento desencadenador: El cliente selecciona un restaurante, eligen un menú disponible y hace clic en el botón realizar pedido.	
Pasos realizados (ruta principal)	Información para los pasos
1. El usuario selecciona un restaurante de la lista de restaurantes disponibles	Página principal de la aplicación, lista de restaurantes
2. El usuario selecciona uno o varios menús de la lista del restaurante y da clic en el botón pedir .	Lista de menús de cada restaurante, dirección de entrega, forma de entrega (recoger pedido, domicilio)
3. Se presenta una ventana de confirmación de pedido con el precio total y el usuario da clic en confirmar pedido.	Venta de confirmación de pedido, precio total de pedido.
Precondiciones: El usuario ha iniciado sesión en la aplicación.	
Requerimientos cumplidos: Permitir que los usuarios puedan realizar pedidos en los distintos restaurantes disponibles.	

Tabla 20.
CU04-Visualizar Restaurantes

Nombre del caso de uso: Visualizar restaurantes	Id Único: CU-04
Versión: 1.0	
Actor(es): Cliente	
Descripción: Permitir que los usuarios clientes puedan ver el listado de todos los restaurantes y menús disponibles.	
Evento desencadenador: El cliente ingresa a la aplicación y puede observar los restaurantes y menús disponibles.	
Pasos realizados (ruta principal)	Información para los pasos
1. El cliente ingresa a la aplicación.	Aplicación
2. El cliente visualiza los restaurantes disponibles en la aplicación y selecciona un restaurante.	Página principal de la aplicación, lista de restaurantes
3. El cliente visualiza los distintos menús del restaurante.	Lista de menús del restaurante
Precondiciones: El usuario ingresó a la aplicación.	
Requerimientos cumplidos: Permitir que los usuarios puedan observar los distintos restaurantes y menús de cada restaurante.	
Excepciones:	
3.1. En caso de que el cliente quiera realizar un pedido debe iniciar sesión.	

Tabla 21.
CU05-Gestionar Cuenta

Nombre del caso de uso: Gestionar cuenta	Id Único: CU-05
Versión: 1.0	
Actor(es): Cliente	
Descripción: Permitir que los usuarios clientes puedan modificar o borrar su cuenta	
Evento desencadenador: El usuario ingresa a la pestaña de configuración y puede modificar o eliminar su cuenta dando clic en el botón Gestionar cuenta.	
Pasos realizados (ruta principal)	Información para los pasos
1. El cliente ingresa a la pestaña de configuración.	Pestaña de configuración de la aplicación
2. El cliente da clic en el botón Gestionar cuenta .	Pestaña de configuración
3. Se muestra una pantalla con información del cliente	Ventana de gestión de cuenta, información cliente
4. El cliente puede modificar su cuenta en el caso de uso Modificar cuenta .	Ventana de gestión de cuenta, información cliente.

5. El cliente puede eliminar su cuenta en el caso de uso Inhabilitar cuenta .	Ventana de gestión de cuenta, información cliente
Precondiciones: El usuario inició sesión en la aplicación e ingresó en la pestaña de configuración.	
Requerimientos cumplidos: Permitir que los clientes puedan modificar o borrar su cuenta.	

Tabla 22.
CU06-Modificar Cuenta

Nombre del caso de uso: Modificar cuenta	Id Único: CU-06
Versión: 1.0	
Actor(es): Cliente	
Descripción: Permitir que los clientes puedan modificar sus cuentas	
Evento desencadenador: El cliente ingresa a la pestaña de configuración y puede modificar su cuenta dando clic en el botón " Modificar cuenta".	
Pasos realizados (ruta principal)	Información para los pasos
1. El cliente ingresa a la pestaña de configuración.	Pestaña de configuración de la aplicación
2. El cliente da clic en el botón Modificar cuenta .	Pestaña de configuración de la aplicación
3. El cliente ingresa información de la cuenta y da clic en el botón Guardar .	Formulario de registro, información del cliente, nombres, apellidos, correo electrónico y contraseña.
4. Se valida la información del cliente	Formulario de registro
5. Se muestra la pestaña de confirmación.	Pestaña de confirmación de la aplicación.
Precondiciones: El usuario inició sesión en la aplicación e ingreso a la pestaña de configuración.	
Requerimientos cumplidos: Permitir que los clientes puedan modificar su cuenta	
Excepciones:	
3.1. El cliente da clic en el botón cancelar	
3.2 El cliente ingresa información no válida.	

Tabla 23.
CU07- Inhabilitar Cuenta

Nombre del caso de uso: Inhabilitar cuenta	Id Único: CU-07
Versión: 1.0	
Actor(es): Cliente	
Descripción: Permitir que los clientes puedan eliminar su cuenta.	
Evento desencadenador: El cliente ingresa a la pestaña de configuración y puede eliminar su cuenta dando clic en el botón " Eliminar cuenta".	
Pasos realizados (ruta principal)	Información para los pasos

1. El cliente ingresa a la pestaña de configuración.	Pestaña de configuración de la aplicación
2. El cliente da clic en el botón Eliminar cuenta .	Pestaña de configuración de la aplicación
3. Se muestra un mensaje de confirmación y el donante da clic en el botón aceptar .	Mensaje de confirmación
4. Se muestra la pestaña de confirmación de la cuenta eliminada.	Pestaña de confirmación de eliminación de cuenta.
Precondiciones: El usuario inició sesión en la aplicación e ingresó a la pestaña de configuración.	
Requerimientos cumplidos: Permitir que los clientes puedan eliminar su cuenta.	
Excepciones: 3.1 En caso de que el usuario de clic en el botón cancelar no se eliminará su cuenta.	

Tabla 24.
CU08- Recibir Notificación

Nombre del caso de uso: Recibir notificación	Id Único: CU-08
Versión: 1.0	
Actor(es): Restaurante	
Descripción: Permitir que el usuario restaurante pueda recibir notificaciones de algún pedido.	
Evento desencadenador: Al registrarse algún pedido a determinado restaurante, este recibirá una notificación en la aplicación.	
Pasos realizados (ruta principal)	Información para los pasos
1. El Cliente realiza un pedido al restaurante.	Aplicación
2. El usuario restaurante recibe una notificación de pedido en la aplicación.	Notificación emergente del pedido del cliente
3. El usuario restaurante da clic en la notificación y ve los detalles del pedido.	Detalles pedidos, menú, precio, nombres cliente, dirección, teléfono.
Precondiciones: El Usuario Restaurante se encuentra logueado en la aplicación.	
Requerimientos cumplidos: Permitir que el usuario restaurante pueda recibir notificaciones de algún pedido.	

Tabla 25.
CU09-Gestionar Catálogo

Nombre del caso de uso: Gestionar catálogo	Id Único: CU-09
Versión: 1.0	
Actor(es): Restaurante	
Descripción: Permitir que el usuario restaurante pueda agregar, modificar o eliminar sus menús.	
Evento desencadenador: El usuario restaurante ingresa a la pestaña de configuración y puede agregar, modificar o eliminar su menú dando clic en el botón Gestionar catálogo.	
Pasos realizados (ruta principal)	Información para los pasos
1. El usuario restaurante ingresa a la pestaña de configuración.	Pestaña de configuración de la aplicación
2. El usuario restaurante da clic en el botón Gestionar catálogo .	Pestaña de configuración
3. El usuario restaurante puede añadir un nuevo menú en el caso de uso Agregar menú .	Ventana de gestión de catálogo, información del menú.
4. El usuario restaurante puede modificar los datos del menú en el caso de uso Modificar menú .	Listado de menús, información del menú.
5. El usuario restaurante puede eliminar algún menú en el caso de uso Eliminar menú .	Listado de menús, información del menú.
Precondiciones: El usuario restaurante inició sesión en la aplicación e ingresó en la pestaña de configuración.	
Requerimientos cumplidos: Permitir que los usuarios restaurante puedan agregar, modificar o eliminar sus menús.	

Tabla 26.
CU10-Agregar Menú

Nombre del caso de uso: Agregar menú	Id Único: CU-10
Versión: 1.0	
Actor(es): Restaurante	
Descripción: Permitir que el usuario restaurante pueda agregar sus menús.	
Evento desencadenador: El usuario restaurante ingresa a la pestaña de configuración y puede agregar menús dando clic en el botón agregar menú.	
Pasos realizados (ruta principal)	Información para los pasos
1. El usuario restaurante ingresa a la pestaña de configuración.	Pestaña de configuración de la aplicación
2. El usuario restaurante da clic en el botón agregar pedidos .	Pestaña de configuración de la aplicación
3. El usuario restaurante selecciona si va a ingresar comida o una oferta, ingresa información del menú y da clic en el botón Agregar .	Formulario de menú, descripción de la comida, descripción de bebida, imagen del menú y precio.

4. Se valida la información del menú.	Formulario de menú
5. Se muestra la pestaña de confirmación.	Pestaña de confirmación de la aplicación.
Precondiciones: El usuario restaurante inició sesión en la aplicación.	
Requerimientos cumplidos: Permitir que el usuario restaurante pueda agregar sus menús	
Excepciones: 3.1 El usuario restaurante ingresa información incorrecta. 3.2 El usuario restaurante da clic en el botón cancelar.	

Tabla 27.
CU11-Modificar Menú

Nombre del caso de uso: Modificar menú	Id Único: CU-11
Versión: 1.0	
Actor(es): Restaurante	
Descripción: Permitir que el usuario restaurante pueda modificar sus menús.	
Evento desencadenador: El usuario restaurante selecciona un menú puede modificar el menú dando clic en el botón modificar.	
Pasos realizados (ruta principal)	Información para los pasos
1. El usuario restaurante ingresa a la aplicación y observa el listado de sus menús.	Listado de los menús
2. El usuario restaurante selecciona un menú y da clic en el botón modificar .	Detalle del menú
3. El usuario restaurante ingresa la información nueva del menú y da clic en el botón Guardar .	Formulario de menú, descripción de la comida, descripción de bebida, imagen del menú y precio.
4. Se valida la información del menú.	Formulario de menú
5. Se muestra la pestaña de confirmación.	Pestaña de confirmación de la aplicación.
Precondiciones: El usuario restaurante inició sesión en la aplicación y existe al menos un menú agregado.	
Requerimientos cumplidos: Permitir que el usuario restaurante pueda modificar sus menús	
Excepciones: 3.1 El usuario restaurante ingresa información incorrecta. 3.2 El usuario restaurante da clic en el botón cancelar.	

Tabla 28.
CU12-Eliminar Menú

Nombre del caso de uso: Eliminar menú	Id Único: CU-12
Versión: 1.0	
Actor(es): Restaurante	
Descripción: Permitir que el usuario restaurante pueda eliminar sus menús.	
Evento desencadenador: El usuario restaurante selecciona un menú y puede eliminar el menú dando clic en el botón eliminar.	
Pasos realizados (ruta principal)	Información para los pasos
1. El usuario restaurante ingresa a la aplicación y observa el listado de sus menús.	Listado de los menús
2. El usuario restaurante selecciona un menú y da clic en el botón eliminar .	Detalle del menú
3. Se muestra un mensaje de confirmación y el usuario da clic en el botón aceptar .	Mensaje de confirmación
4. Se muestra la pestaña de confirmación del menú eliminado.	Pestaña de confirmación de eliminación de menú.
Precondiciones: El usuario restaurante inició sesión en la aplicación y existe al menos un menú agregado.	
Requerimientos cumplidos: Permitir que el usuario restaurante pueda eliminar sus menús.	
Excepciones:	
3.1 El usuario restaurante da clic en el botón cancelar.	

Prototipado de la app versión 1.0

Para la elaboración de los prototipos se hizo uso de la Figma³, la cual es una herramienta que permitió al equipo de desarrollo, crear los prototipos de la aplicación móvil. Esta herramienta mostraba las interfaces de usuario, lo cual permitió establecer si era o no lo que el cliente quería.

Al utilizar la herramienta antes mencionada, el equipo creó la primera versión de la aplicación, quedando de la siguiente manera:

Las pantallas de inicio, registro, inicio de sesión y formulario de inicio de sesión, se muestran en la Figura 17.

³ <https://www.figma.com>

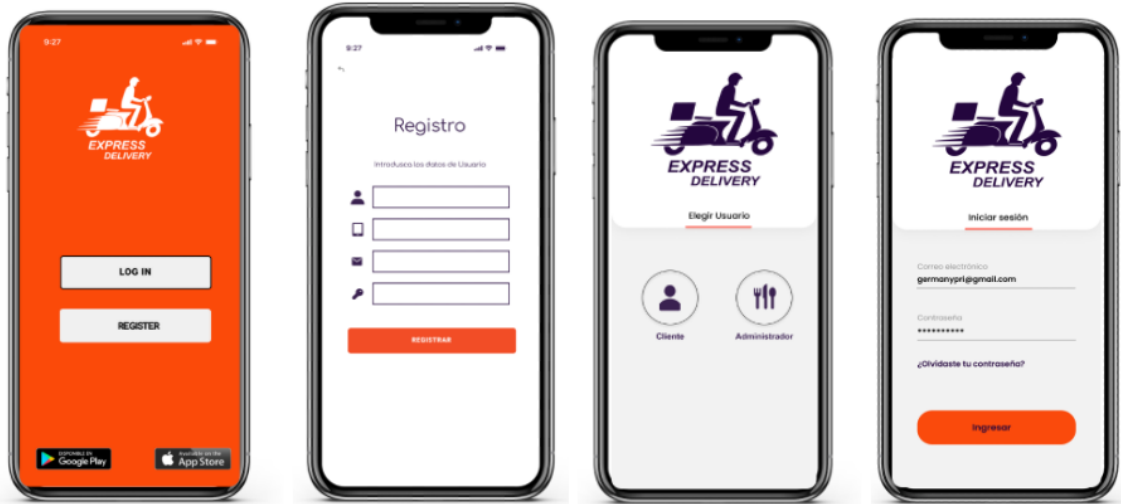


Figura 17. Prototipo v1.0-Registro y Autenticación

Las pantallas del menú principal, lista de restaurantes, perfil de cliente y los pedidos, se muestran en la Figura 18.



Figura 18. Prototipo V1.0-Menú principal

Las pantallas de descripción del producto, carrito del pedido, pedido exitoso y agregar producto por parte del restaurante, se muestran en la Figura 19.

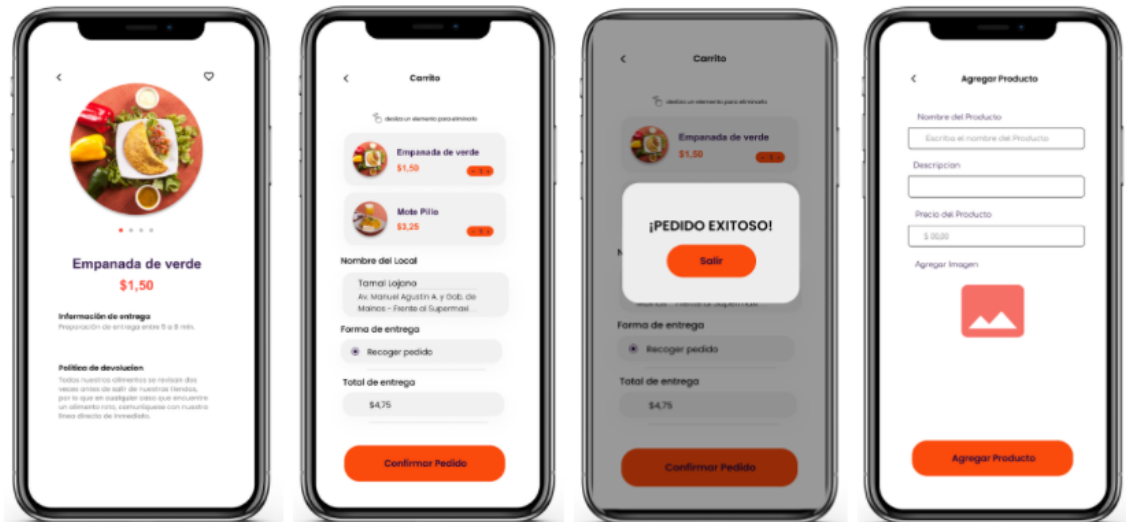


Figura 19. Prototipo v1.0-Realizar pedidos

Casos de Uso del Prototipo versión 2.0

Al establecer una reunión entre todos los miembros del equipo para la presentación del Prototipo v1.0, el cliente sugirió algunos cambios, por lo que se tuvo que modificar el diagrama de casos de uso general. Este cambio fue el de agregar un apartado de promociones que ofrecen el restaurante, como se puede observar en la Figura 20.



Figura 20. Casos de Uso del Prototipo v2.0

Descripción del Caso de Uso del Prototipo versión 2.0

Tabla 29.
CU13-Ver promociones

Nombre del caso de uso: Ver promociones		Id Único: CU-13
Versión: 1.0		
Actor(es): Cliente		
Descripción: Permitir que el cliente pueda observar menús que están en promociones en determinado tiempo.		
Evento desencadenador: El cliente ingresa a la interfaz de promociones y puede observar los menús que están en promoción.		
Pasos realizados (ruta principal)		Información para los pasos
1. El cliente ingresa al apartado de promociones		Interfaz de promociones
2. El cliente puede visualizar los menús que están en promoción en ese momento.		Lista de menús en promoción.
3. El cliente puede seleccionar una promoción de su interés y ver a qué restaurante pertenece.		Menús del restaurante
Precondiciones: El cliente inició sesión en la aplicación.		
Requerimientos cumplidos: Permitir que el cliente pueda observar menús que están en promociones en determinado tiempo.		

Prototipado de la app versión 2.0

Una vez establecidos los cambios, se procedió a construir la segunda versión del prototipo, como se observa en la Figura 21. Entre los cambios aplicados para esta nueva versión son:

- Nombre de la aplicación RAPINASA
- Cambio en el logo de la aplicación, debido a que no se trataba precisamente de un servicio de delivery, sino de una aplicación dedicada a realizar pedidos de comida, lo cual al final el cliente decidía si desea recoger el pedido o que se le envié a domicilio, pero esto es un proceso externo.
- Se agregó la calificación de los restaurantes visualizada en la pantalla de inicio de restaurantes.
- En la vista del carrito se agregó las opciones de entrega a domicilio o recoger pedido.
- En el menú desplegable se hicieron las siguientes modificaciones:
 - Eliminación de la opción seguridad

- Modificación del nombre de la pestaña “ofertas y promoción” a simplemente “promoción”.
- Se agregó historial de pedidos
- Se eliminaron las opciones de preguntas frecuentes y ayuda del perfil de usuario del cliente.

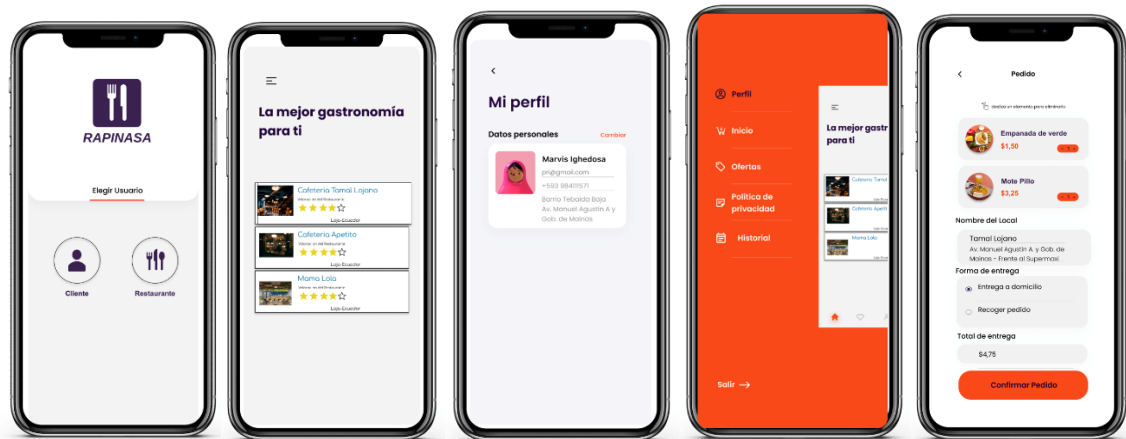


Figura 21. Prototipo de la app de pedidos v2.0

Además, se agregaron nuevas vistas en esta versión 2.0, estas se describen a continuación:

- Se agregó un menú correspondiente a la sección de restaurantes, donde pueden agregar las promociones y productos que ofrecen, esto se puede observar en la Figura 22.

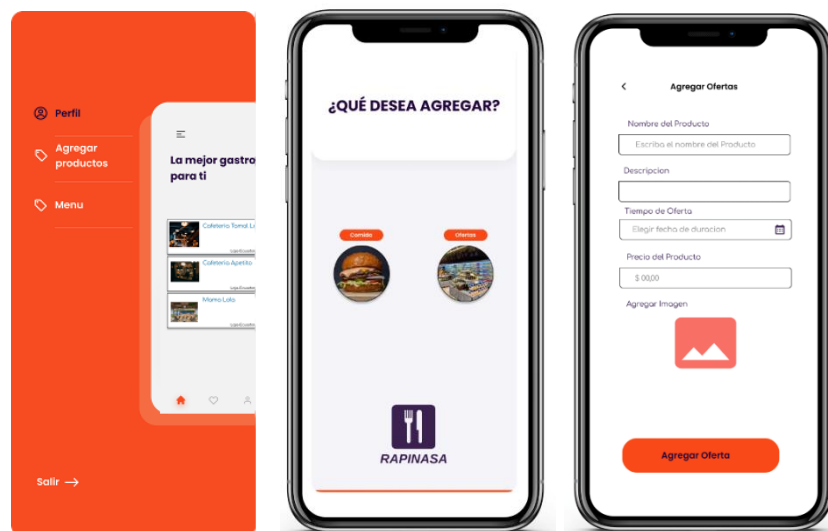


Figura 22. Prototipo v2.0-menu desplegable para productos y promociones

- También se agregaron la sección de historial de pedidos, políticas y privacidad y el perfil del restaurante, como se observa en la Figura 23. Eso fue propuesto por el equipo de desarrollo y aceptado por cliente.



Figura 23. Prototipo v2.0- Políticas, historial y perfil del restaurante

Fase de desarrollo

Para el desarrollo de la aplicación de pedidos de comida, el equipo de desarrollo decidió crear la app para dispositivos Android ajustándose a los requerimientos del cliente. Por ello, para la programación hicieron uso de IDE Android Studio donde programaron todas las funcionalidades de la aplicación.

Para la gestión de los datos, incluidas las imágenes para la app, se hizo uso de Firebase la cual es una plataforma en la nube donde se pudo generar la base de datos NoSql utilizada en la aplicación. En la Figura 24, se muestra la estructura de la Base de Datos en Firebase, donde se pueden ver las diferentes clases que el equipo de desarrollo suscitó para la aplicación móvil de pedidos de comida a domicilio.

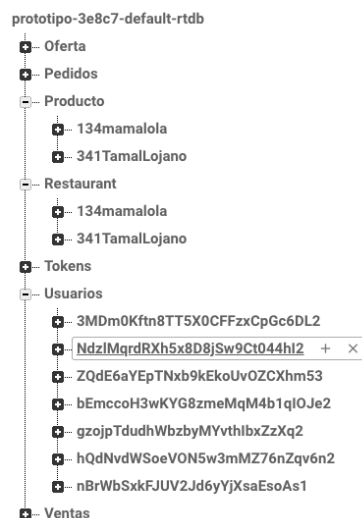


Figura 24. Base de Datos de la app de pedidos

Finalmente, al terminar el desarrollo de la app, el equipo compartió el código fuente a todos los miembros del equipo para que por consiguiente se realicen las pruebas respectivas, de acuerdo a la organización del equipo.

El repositorio donde se encuentra el código fuente de la aplicación se encuentra en el siguiente enlace:

<https://drive.google.com/drive/folders/1mhyDRt-YWVMJFjU-eOglaDMHzYdfmqFj>

Fase de pruebas

El equipo de desarrollo planificó realizar pruebas de escenarios y pruebas de aceptación (**ver Anexo 2 y 3**), para lo cual se ha identificado los diferentes casos de pruebas donde se registran los módulos correspondientes para cumplir con dichos escenarios, además de especificar los requerimientos pertinentes para llevar a cabo el módulo y una descripción del mismo; resultando con un total de 15 escenarios identificados como se pueden observar en la Figura 25.

ID	Escenario	Fecha de Creación	tipo	Modulo	Requerimiento	Detalle del proceso
1	Registro de un usuario en el sistema		post	RegistrarUsuario	nombre, apellidos, dirección, teléfono, correo electrónico, contraseña.	Permitir que los usuarios clientes se registren para que puedan tener su cuenta.
2	Ingreso de un usuario en el sistema		post	IdentificarUsuario	Correo, password, tipo	Permite que los usuarios clientes y usuarios restaurantes puedan iniciar sesión en la aplicación.
3	Realizar pedido por un cliente		post	RealizarPedido	producto, precio, cantidad, dirección, tipo, entrada	Permitir que los clientes puedan realizar pedidos a los restaurantes disponibles en la aplicación.
4	Mostrar los restaurantes disponibles		get	VisualizarRestaurantes	nombreRestaurante, valoración, dirección	Permitir que los usuarios clientes puedan ver el listado de todos los restaurantes y menús disponibles.
5	Modificar datos de una cuenta cliente		post	ModificarCuenta	nombre, correo, teléfono, dirección, foto, perfil	El cliente ingresa a su perfil y puede modificar su cuenta dando clic en el botón "Modificar cuenta"
6	Mostrar historial de pedidos		get	HistorialPedidos	nombreRestaurante, totalPedido, dirección, cantidad, productos	El usuario cliente ingresa a la pestaña historial y puede observar el historial de sus pedidos.
8	Recibir notificación sobre un pedido		get	RecibirNotificacion	detalles pedido, menú, precio, nombres cliente, dirección, teléfono.	Al registrarse algún pedido a determinado restaurante, éste recibirá una notificación en la aplicación.
9	Gestion de los productos de los restaurantes		post	GestionarCatalogo		El usuario restaurante ingresa a la pestaña de agregar productos y puede agregar, modificar o eliminar sus menús.
10	Gestion de las ofertas de los restaurantes		post	GestionarCatalogo		El usuario restaurante ingresa a la pestaña de agregar ofertas y puede agregar, modificar o eliminar sus ofertas.
11	Agregar un menú al catálogo del restaurante		post	AgregarProducto	nombreProducto, descripciónProducto, precioProducto, imagenProducto	El usuario restaurante ingresa a la pestaña de agregar producto y puede agregar productos dando clic en el ícono comida.
12	Modificar menú del catálogo del restaurante		post	ModificarProducto	nombreProducto, descripciónProducto, precioProducto, imagenProducto (no es necesario modificar todos los campos)	El usuario restaurante selecciona un producto puede modificar el producto dando clic en el botón modificar.
13	Agregar oferta de restaurante		post	AgregarOferta	nombreOferta, descripciónOferta, precioOferta, imagenOferta, tiempoOferta	El usuario restaurante ingresa a la pestaña de agregar oferta y puede agregar ofertas dando clic en el ícono oferta.
14	Modificar oferta del restaurante		post	ModificarOferta	nombreOferta, descripciónOferta, precioOferta, imagenOferta, tiempoOferta	El usuario restaurante selecciona un producto puede modificar el producto dando clic en el botón modificar.
15	Gestion de los restaurantes por parte del administrador		post	GestionarRestaurante	nombreRestaurante, valoración, dirección, teléfono, correo, imagenRestaurante	El administrador agrega los restaurantes desde la base de datos directamente.

Figura 25. Pruebas de escenarios Fase I

Luego de identificar los diferentes escenarios se procede a evaluar los mismos comparando los resultados esperados con los resultados obtenidos del testeo, esto con el fin de comprobar la funcionalidad del aplicativo móvil en su totalidad, a continuación, se puede evidenciar los resultados (**ver Anexo 2**), en donde se garantiza la funcionalidad de la app satisfactoriamente, esto se puede observar en la Figura 26.

ID	Modulo	Requerimiento	Detalle del proceso	Resultado esperado	Ejecución de pruebas	
					Fecha	Resultado
1	RegistrarUsuario	nombre, apellidos, dirección, teléfono, correo electrónico, contraseña.	Permitir que los usuarios clientes se registren para que puedan tener su cuenta.	el usuario ingresado se registrara correctamente	2021-08-02	usuario registrado correctamente
2	RegistrarUsuario	nombre, apellidos, dirección, teléfono, correo electrónico, contraseña.	Se tratara de registrarse con el mismo correo nuevamente	el usuario ingresado ya esta registrado	2021-08-02	mensaje de usuario ya registrado
3	RegistrarUsuario	nombre, apellidos, dirección, teléfono, correo electrónico, contraseña.	No se llenara todos los campos del formulario	No se puede crear la cuenta debido a los campos vacios	2021-08-02	mensaje de usuario no registrado
4	IdentificarUsuario	Correo, password	Permite que los usuarios clientes puedan iniciar sesión en la aplicación.	Ingreso a la aplicacion movil exitosa	2021-08-02	ingreso exitoso
5	IdentificarUsuario	Correo, password,	Permite que los usuarios restaurante puedan iniciar sesión en la aplicación.	Ingreso a la aplicacion movil exitosa	2021-08-02	ingreso exitoso
6	RealizarPedido	producto,precio,cantidad,direccion,tipoEntrega	Permitir que los clientes puedan realizar pedidos a los restaurantes disponibles en la aplicación.	Pedido realizado con exito	2021-08-02	Pedido registrado exitosamente
7	VisualizarRestaurantes	nombreRestaurante,valoracion,direccion	Permitir que los usuarios clientes puedan ver el listado de todos los restaurantes y menús disponibles.	Se debe visualizar todos los restaurantes registrados en la pantalla de inicio	2021-08-02	Los restaurantes se pueden visualizar exitosamente

Figura 26. Pruebas escenarios Fase II

Finalmente, el equipo realizó una encuesta dirigida al cliente (autor del TT), con la finalidad de verificar el nivel de aceptación del mismo y por ende recibir la aceptación final del proyecto, logrando así el objetivo de desarrollar una aplicación móvil para realizar pedidos de comida **(ver Anexo 3)**.

Reuniones

En la metodología de desarrollo de software por prototipos, es importante mantener las reuniones con el cliente, esto se realizó con la finalidad de cumplir todas las expectativas del mismo; por tal motivo se establecieron reuniones entre todos los miembros del equipo presentando los avances y las versiones de la aplicación. Las reuniones planificadas se muestran en la Tabla 30, así también se adjunta como evidencia el link de los videos de las mismas.

**Tabla 30.
Reuniones-Prototipado**

Fecha	Motivo	Duración (min)	Entregable	URL
2020/06/30	inicio del proyecto	30		https://youtu.be/gyGUECzF9no
2020/07/05	elicitación de requerimientos	15		https://youtu.be/VkyQxFmmhU
2020/07/09	Presentación de Casos de Uso	10		https://youtu.be/aAMwx8aYEfM
2020/07/18	Presentación del prototipo V.1	20	Prototipo V 1.0	https://youtu.be/niOSMQLDRJo
2020/07/19	Cambios requeridos	20		https://youtu.be/XbC_xGG0ix8
2020/07/25	Presentación del prototipo V.2	10	Prototipo V 1.2	https://youtu.be/M1AFoWeSsHw
2020/07/30	Avances en el desarrollo de la aplicación	5		https://youtu.be/3-n4qrHepVI
2020/08/05	Presentación y entrega de la aplicación	20	Producto Final	https://youtu.be/jJ06uaKtX9Y

Recopilación de información.

Luego de haber concluido la fase de operación en ambos escenarios, se procedió a tabular en una tabla comparativa la información obtenida durante el proceso de acuerdo a las tareas asignadas por cada equipo. Estos datos se muestran en la Tabla 31.

Tabla 31.
Información de la experimentación

ID	Descripción	Escenario: PROTOTIPADO
T01	Realizar una aplicación móvil para realizar pedidos de comida.	Se desarrolló una sola aplicación para gestionar todos los componentes del sistema.
T02	La aplicación debe ser desarrollada para dispositivos Android.	El equipo desarrolló una aplicación únicamente para dispositivos Android, programando en Android Nativo.
T03	Se deben aplicar todas las fases de la metodología correspondiente.	Fases aplicadas: <ul style="list-style-type: none">• Análisis• Diseño• Construcción del prototipo• Desarrollo del prototipo• Pruebas
T04	Documentar todos los procesos realizados durante la creación de la aplicación.	La documentación moderada
T05	Cumplir con el tiempo de entrega (tiempo max. 5 semanas).	Cumplió con el tiempo establecido.
T06	Subir el código fuente a un repositorio.	https://drive.google.com/drive/folders/1mhyDRt-YWVMJFjU-eOglaDMHzYdfmqFj
T07	Reunirse con el cliente por la menos 1 vez por semana.	Se establecieron reuniones para cada entregable del prototipo.

6.3. Objetivo 3: Evaluar las metodologías de desarrollo de software para aplicaciones móviles de acuerdo a los criterios de evaluación establecidos.

En este objetivo, se continúa con la siguiente etapa del proceso experimental definido en el Objetivo 2, esta es: Análisis e interpretación y Presentación de resultados (ver Figura 27). Para ello, se realizó la evaluación de la metodología Prototipado utilizando la matriz diseñada en el objetivo 1 (ver sección Resultados, Objetivo 1), obteniendo datos que permitieron realizar la evaluación y comprobar la hipótesis planteada. También se determinó fortalezas y debilidades de la metodología aplicada.

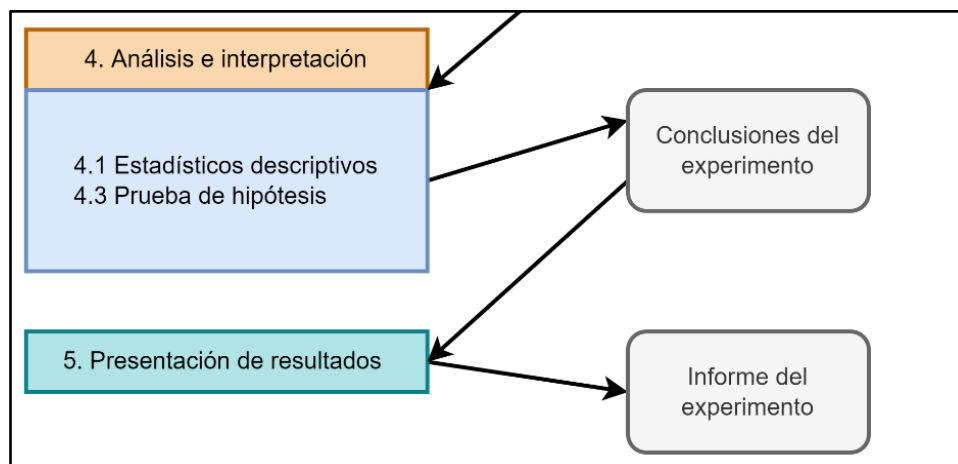


Figura 27. Fase II del proceso experimental

Análisis e interpretación

Para obtener los datos estadísticos, se efectuó la comparativa entre Scrum y Prototipado, empleando la matriz creada en el Objetivo 1, por ende, se procedió evaluar los criterios definidos colocándoles un valor booleano (Verdadero o Falso).

Para observar de una mejor manera la calificación, se procedió a evaluar por cada categoría definida en dicha matriz con base en todos los resultados presentados en el Objetivo 2.

Categoría 1: Usabilidad

Tabla 32.
Evaluación por categoría-Usabilidad

Usabilidad	Descripción	V-F
Adaptación a entornos turbulentos	La metodología se adapta a una forma de trabajo poco común y con riesgos que pueden afectar el proyecto.	V
Satisfacción del cliente	El cliente se encuentra satisfecho con el producto final.	V
Estándares de calidad	La metodología aplicada se basa en estándares que otorguen más calidad al producto.	F
Respeto de las fechas de entregas	Se cumple con las fechas de entrega del producto.	V
Cumplimiento de requerimientos	Se cumple con todos los requerimientos establecidos por el cliente, haciéndose mediante pruebas.	V
Trabajo remoto	La metodología se adapta a la modalidad de trabajo remoto.	V
Herramientas	Existen múltiples herramientas que faciliten los procesos que conlleva la metodología.	V

Categoría 2: Aplicabilidad

Tabla 33.
Evaluación por categoría-**Aplicabilidad**

Aplicabilidad	Descripción	V-F
Interacción entre los miembros del equipo	La metodología permite que los miembros del equipo interactúen entre sí, con el fin de ejecutar el proyecto de la mejor manera.	V
Interacción con el cliente	La metodología permite que algunos o todos los miembros del equipo interactúen con el cliente, esto con el objetivo de entregar un producto de calidad y cumpliendo con todos los requerimientos al cliente.	V
Evaluación de riesgos	Los miembros del equipo desarrollan un plan de riesgos para evaluar.	F
Aplicable para proyectos pequeños	Es posible aplicar la metodología seleccionada en proyectos que no requieran muchos recursos, ni tiempo.	V
Requiere una buena organización del equipo.	La metodología permite establecer roles y organizar las tareas para cada uno de estos.	V
Aplicable en equipos pequeños	Se puede aplicar la metodología con un equipo pequeño de 5 personas.	V
Adaptable a cambios	La metodología se adapta sin ningún problema a cambios requeridos por el cliente.	V

Basados en casos de uso	Los miembros del equipo utilizan casos de uso para describir de una mejor manera los requerimientos.	V
Documentación adecuada	Los miembros del equipo documentan todos los procesos ejecutados durante el desarrollo del proyecto.	V
Integración entre las etapas de desarrollo.	La metodología permite conectar o unificar sus fases para facilitar los procesos.	V
Relación con UML	Los miembros del equipo utilizan UML como lenguaje para modelar la arquitectura, diseño e implementación del sistema.	V
Permite el desarrollo de software para aplicaciones móviles	La metodología permite adaptarse a cualquier tipo de proyecto de desarrollo de software en aplicaciones móviles.	V
Aplicable en tiempos pandémicos	Se puede aplicar la metodología bajo efectos y restricciones de la pandemia del covid-19.	V

Categoría 3: Procesos y Productos

Tabla 34.
Evaluación por categoría-Procesos y Productos

Procesos y Productos	Descripción	V-F
Pruebas unitarias o de escenarios	El equipo de desarrollo realiza pruebas unitarias o de escenarios.	V
Pruebas de aceptación	El equipo de desarrollo realiza pruebas de aceptación en conjunto con el cliente.	V
Control de Calidad	El equipo de desarrollo realiza un seguimiento en la calidad del producto.	V
Producto funcional /Ejecutable	El equipo de desarrollo entrega un producto funcional al cliente.	V

Categoría 4: Capacidad

Tabla 35.
Evaluación por categoría-Capacidad

Capacidad	Descripción	V-F
Los requisitos funcionales pueden cambiar	La metodología permite cambiar los requisitos funcionales obtenidos durante el análisis de requerimientos.	V
Requisitos no funcionales pueden cambiar	La metodología permite cambiar los requisitos no funcionales.	V
Roles intercambiables	La metodología aplicada es flexible en la organización del equipo, permitiendo cambiar de roles y por ende tareas.	F

Integración de los cambios	La metodología permite realizar cambios sin que afecte a los procesos de las demás fases de la metodología aplicada.	V
Conocimiento compartido	Existe un conocimiento compartido entre los miembros del equipo, lo cual disminuye la incertidumbre en la ejecución del proyecto.	V
Colaboración	La metodología permite que exista una colaboración en la ejecución de las tareas que tiene cada miembro del equipo.	V
Ambiente de trabajo favorable	La metodología asegura de que exista un buen ambiente de trabajo entre los miembros del equipo y el cliente.	V
Posibilidad de reactividad (cada etapa, cada iteración)	La metodología permite fácilmente reactivar o reprogramar las actividades en cada Sprint.	V
Iteraciones cortas	La metodología tiene ciclos pequeños que no llevan más de 4 semanas.	V
El plan de trabajo se puede cambiar	El plan de trabajo establecido se puede cambiar fácilmente.	V

Valoración de los criterios

Luego de evaluar se procedió a unificar las categorías en una sola matriz, con sus parámetros ya evaluados, como se puede observar en la Tabla 36.

Tabla 36.
Comparativa Experimental

Categorías	Métricas	Metodología de Desarrollo
		PROTOTIPADO
USABILIDAD	Adaptarse a entornos turbulentos	V
	Satisfacción del cliente	V
	Estándares de calidad	F
	Respeto de las fechas de entregas	V
	Cumplimiento de requerimientos	V
	Trabajo remoto	V
	Herramientas	V
APLICABILIDAD	Interacción entre los miembros el equipo	V
	Interacción con el cliente	V
	Evaluación de riesgos del proyecto	F
	Aplicable para proyectos pequeños	V
	Requiere una buena organización del equipo	V
	Aplicable en equipos pequeños	V

	Adaptable a cambios	V
	Basados en casos de uso	V
	Documentación adecuada	V
	Integración entre las etapas de desarrollo	V
	Relación con UML	V
	Permite el desarrollo de software para aplicaciones móviles	V
	Aplicable en tiempos pandémicos	V
PROCESOS Y PRODUCTOS	Pruebas unitarias o de escenarios	V
	Pruebas de aceptación	V
	Control de Calidad	V
	Ejecutable	V
CAPACIDAD	Los requisitos funcionales pueden cambiar	V
	Requisitos no funcionales pueden cambiar	V
	Roles intercambiables	F
	Integración de los cambios	V
	Conocimiento compartido	V
	Colaboración	V
	Ambiente de trabajo favorable	V
	Posibilidad de reactividad (cada etapa, cada iteración)	V
	Iteraciones cortas	V
	El plan de trabajo se puede cambiar	V

Estadísticos Descriptivos

Para obtener resultados con datos cuantitativos, y de esta manera realizar el análisis estadístico; se procedió a convertir los valores de **Verdadero = 1**, y **Falso = 0**. Esto se muestra en la Tabla 37.

Tabla 37.
Evaluación Experimental

Categorías	Métricas	Metodología de Desarrollo
		PROTOTIPADO
USABILIDAD	Adaptarse a entornos turbulentos	1
	Satisfacción del cliente	1
	Estándares de calidad	0
	Respeto de las fechas de entregas	1
	Cumplimiento de requerimientos	1
	Trabajo remoto	1
	Herramientas	1
APLICABILIDAD	Interacción entre los miembros el equipo	1
	Interacción con el cliente	1
	Evaluación de riesgos del proyecto	0
	Aplicable para proyectos pequeños	1
	Requiere una buena organización del equipo	1
	Aplicable en equipos pequeños	1

	Adaptable a cambios	1
	Basados en casos de uso	1
	Documentación adecuada	1
	Integración entre las etapas de desarrollo	1
	Relación con UML	1
	Permite el desarrollo de software para aplicaciones móviles	1
	Aplicable en tiempos pandémicos	1
PROCESOS Y PRODUCTOS	Pruebas unitarias o de escenarios	1
	Pruebas de aceptación	1
	Control de Calidad	1
	Ejecutable	1
CAPACIDAD	Los requisitos funcionales pueden cambiar	1
	Requisitos no funcionales pueden cambiar	1
	Roles intercambiables	0
	Integración de los cambios	1
	Conocimiento compartido	1
	Colaboración	1
	Centrado en las personas	1
	Posibilidad de reactividad (cada etapa, cada iteración)	1
	Iteraciones cortas	1
	El plan de trabajo se puede cambiar	1
TOTAL		31
		91%

La Figura 28 muestra los resultados de la comparación experimental, se puede observar la puntuación obtenida por cada categoría evaluada. Los datos de la gráfica se muestran en la Tabla 38.

Tabla 38.
Resultados de la Evaluación por Categorías

CATEGORIAS	PUNTAJE
USABILIDAD	6/7
APLICABILIDAD	12/13
PROCESOS Y PRODUCTOS	4/4
CAPACIDAD	9/10
TOTAL	31/34

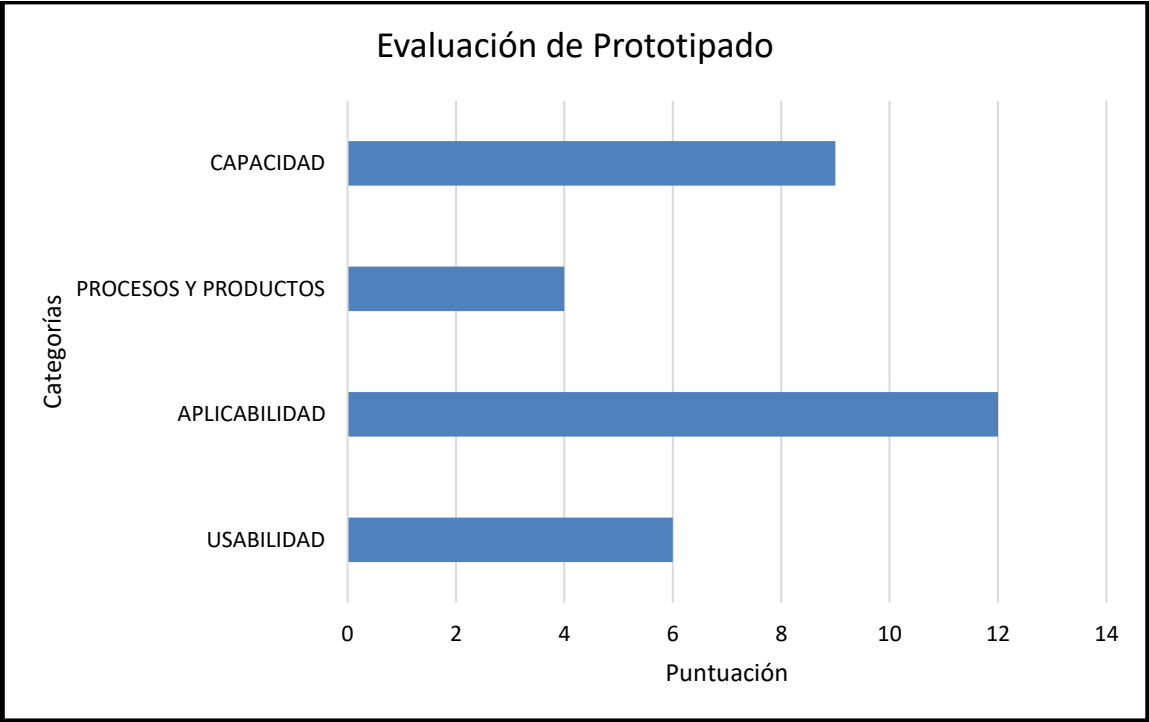


Figura 28. Resultados de la comparación experimental

Comprobación de la Hipótesis

Para la comprobación de la hipótesis se formula la hipótesis de investigación y la hipótesis nula:

- **H_i**: Prototipado es una metodología de desarrollo de software que se adapta al desarrollo de aplicaciones móviles en tiempos de pandemia del covid-19.
- **H₀**: Prototipado es una metodología de desarrollo de software que no se adapta al desarrollo de aplicaciones móviles en tiempos de pandemia del covid-19.

En la Tabla 39, se indica la escala de aceptación de la metodología, la cual sirvió para rechazar o aceptar la hipótesis, este modelo de medición se lo adecuó para el presente Trabajo de Titulación [35].

Tabla 39.
Niveles de puntuación

Escala de medición (%)	Niveles de puntuación	Grado de satisfacción	Premisas
80 – 100	Cumple con las expectativas	Muy satisfactorio	• Puntuación Final <60%, se acepta H₀ • Puntuación Final >60%, se acepta H_i
60 – 79	Aceptable	Satisfactorio	
40 – 59	Poco aceptable	Insatisfactorio	
0 – 39	Inaceptable		

De acuerdo a las premisas, se rechaza **H₀** y acepta **H_i** debido a que la Puntuación Final fue del **91%**.

Presentación de Resultados

Luego de finalizar el proceso experimental aplicando todas las fases del mismo, se obtuvo que Prototipado obtuvo un grado de satisfacción del 91%, cumpliendo con las expectativas de la experimentación, por lo tanto, se acepta la hipótesis de investigación:

H_i: Prototipado es una metodología de desarrollo de software que se adaptó al desarrollo de aplicaciones móviles en tiempos de pandemia del covid-19.

Definir fortalezas y debilidades de la metodología evaluada.

Luego de realizar el proceso de experimental, y evaluar con los respectivos criterios, se procedió a determinar fortalezas y debilidades con base en la experimentación. Esta determinación se muestra en la Tabla 40.

Tabla 40.
Fortalezas y Debilidades de Prototipado

PROTOTIPADO	
FORTALEZAS	DEBILIDADES
<ul style="list-style-type: none">✓ El desarrollo de software por prototipos es rápido, por lo que se puede obtener un entregable en poco tiempo y cumpliendo con los requerimientos del cliente✓ Los prototipos pueden evolucionar y mejorar sus características, mejorando los requerimientos iniciales.✓ Existen múltiples herramientas para la construcción de prototipos de software, y algunas de estas son gratis, por lo que tanto el equipo como el cliente puede llevarse un ahorro en el presupuesto.✓ Se puede implementar para proyectos pequeños como aplicaciones móviles estándar.✓ Se adapta fácilmente al trabajo remoto y a entornos turbulentos.	<ul style="list-style-type: none">✗ La aplicación de estándares de calidad es casi nula, debido a que el foco principal es la construcción del prototipo en el menor tiempo posible.✗ El prototipo entregado en cada Sprint no siempre cumple con las expectativas del cliente, por lo que se corre el riesgo de que este sea desechado.

7. Discusión

En este apartado se indica el desarrollo a la propuesta alternativa, en donde se presentan los resultados obtenidos del cumplimiento de cada objetivo del presente Trabajo de Titulación, esto desde el punto de vista del presente autor, además se presenta la valoración técnica, económica y ambiental.

7.1. Desarrollo de la propuesta alternativa

El presente TT, estuvo conformado por 3 fases. Estas fases corresponden a cada uno de los objetivos planteados para desarrollar este Trabajo de Titulación. Todas estas fases, actividades y resultados obtenidos se muestran a continuación.

Objetivo 1: Determinar las metodologías de desarrollo de software para aplicaciones móviles a evaluar y establecer los criterios de comparación.

Para realizar este objetivo con éxito, se hizo uso de una Revisión Sistemática de Literatura (RSL) siguiendo la metodología planteada por Bárbara Kitchenham, la cual estuvo enfocada en las metodologías de desarrollo de software más utilizadas durante la pandemia del covid-19 para la creación de aplicaciones móviles. Esta revisión fue tomada como base para determinar las metodologías a evaluar, que a diferencia de los trabajos [1], [5], [22], [23], [24], [25], estos simplemente efectúan una revisión bibliográfica donde seleccionan la metodología a su propio criterio. Para construir la matriz comparativa de acuerdo a los criterios de comparación establecidos (**ver Objetivo 1, Tarea 2**). Para la realización de esta RSL, se planteó 3 preguntas de investigación donde se consultó en 4 bases de datos científicas (ACM, IEEE, Web of Science, Scopus) por medio de cadenas de búsqueda usando operadores lógicos AND y OR, mientras que en los trabajos [5], [22], [23], [24] obtienen información de fuentes grises. Como resultado de la RSL, se obtuvo un total de 308 artículos, de los cuales, aplicando los criterios de inclusión y exclusión, se obtuvieron un total de 24 artículos seleccionados para el desarrollo de la investigación, los cuales servirían para seleccionar las metodologías a comparar. De acuerdo a la primera pregunta de la RSL, Scrum y Prototipado fueron las metodologías más usadas durante la pandemia del covid-19, pero para la evaluación Scrum fue descartado debido a que no cumplía con los conceptos fundamentales de una metodología de desarrollo de software, por ello, solo se seleccionó a Prototipado como la metodología a evaluarse, a diferencia de los trabajos [5] [22], definen a Scrum como una metodología de desarrollo de software y no como un framework. Luego se procedió a construir una matriz de evaluación siguiendo una guía comparativa (**ver Objetivo 1, Tarea 2**), donde se propuso métricas de evaluación con el fin de valorar la metodología seleccionada(Prototipado), en cambio, en los trabajos [1], [5], [23], [24], [25] diseñan su propia

matriz de evaluación comparando más de una metodología de desarrollo de software.

Objetivo 2: Desarrollar un entorno de experimentación para la evaluación de las metodologías de desarrollo de software para aplicaciones móviles.

Durante esta fase se procedió a desarrollar un entorno de experimentación con Prototipado. Esto se realizó siguiendo un proceso experimental donde se seleccionó un grupo de sujetos, los cuales tuvieron que desarrollar una aplicación móvil de prueba. Los sujetos seleccionados fueron estudiantes de noveno ciclo de la carrera de Ingeniería en Sistemas y con base en su malla curricular se los consideró aptos para la experimentación. El presente TT, se diferencia de los trabajos [1], [5], [22], [23], [24], [25], debido a que no siguen un proceso experimental estructurado trabajando con un grupo de sujetos reales, y todas las fases de las metodologías a comparar las desarrollan sus propios autores, además de que no son trabajadas bajo entornos turbulentos.

A estos sujetos seleccionados, se les indicó los requerimientos de la aplicación por lo que aplicaron la metodología asignada donde tenían que cumplir con su ciclo de vida y respetar el método de trabajo de la metodología asignada.

El proceso experimental se dividió en dos fases; en este objetivo se trabajó con la primera donde definió el alcance, se hizo la planificación del experimento y la operación, donde se planteó la hipótesis y se recolectó la información necesaria para la evaluación.

En la operación los sujetos tuvieron que desarrollar la aplicación móvil denominada “RapiNasa” de acuerdo a los parámetros establecidos en el grupo, dando como resultado un producto funcional, es decir, una aplicación móvil cumpliendo con los requerimientos del cliente (**ver sección Resultados, Objetivo 2**).

Todo el plan de trabajo, se lo ejecutó remotamente, respetando la integridad y salud de los sujetos, tomando en cuenta la crisis sanitaria por la pandemia del covid-19, que a diferencia de los trabajos [22], [23] llevan un modelo presencial y los trabajos [1], [5], no está claro la modalidad de trabajo.

Objetivo 3: Evaluar las metodologías de desarrollo de software para aplicaciones móviles de acuerdo a los criterios de evaluación establecidos.

Para cumplir con este objetivo se continuó con la segunda parte del proceso experimental “Análisis e interpretación” y “Presentación de Resultados”, para lo cual se hicieron los respectivos análisis estadísticos a partir de la evaluación de Prototipado. Por ello se evaluó con ayuda de la matriz comparativa de acuerdo a los criterios establecidos donde se procedió a calificar de forma binaria si cumplía o no dichos criterios, mientras que en los trabajos [1], [5], [22], [23], [24], [25] trabajan con diferentes valoraciones y grados de satisfacción de acuerdo al criterio del autor.

Como segundo punto, se procedió a sumar los puntajes y obtener porcentajes, dando como resultado a Prototipado con una valoración del 91%, con un grado muy aceptable y cumpliendo con las expectativas planteadas; por lo tanto, se pudo comprobar la hipótesis de investigación y se respondió a la pregunta de investigación del presente Trabajo de Titulación. Finalmente, se pudo definir fortalezas y debilidades de Prototipado basándose en todo el proceso investigativo y experimental, con el fin de establecer ciertas recomendaciones para trabajos futuros o para lectores del presente trabajo de Titulación.

7.2. Valoración técnica, económica, ambiental.

Para el desarrollo del presente trabajo de titulación se hizo uso de recursos técnicos, económicos y ambientales.

Los recursos técnicos permitieron el desarrollo del entorno de experimentación para la de la metodología Prototipado, la cual se evaluó mediante una aplicación móvil, donde se emplearon varias herramientas como frameworks y lenguajes de programación. Los gastos presentados fueron mínimos, debido a que las herramientas utilizadas estuvieron disponibles de forma gratuita, facilitando el estudio comparativo.

Se contribuye positivamente al ambiente, por el hecho de que se trabajó 100% de manera virtual por lo que el uso del papel fue casi nulo.

A continuación, se puede observar de forma detallada los recursos utilizados y el valor económicos de los mismos. En la Tabla 41, se puede muestra el valor económico perteneciente al Talento Humano; los gastos del director y el asesor académico son asumidos por la Universidad Nacional de Loja por lo que no se considerarán gastos de forma directa.

Tabla 41.
Valoración Económica-Talento Humano

Nombre	Cargo	Número de horas	Precio/Hora	(\$) Valor Total
Andy Aron Camacho Herrera	Tesista	400	\$8.00	\$3200.00
Edison Leonardo Coronel Romero	Director	200	\$0.00	\$0.00
Francisco Javier Álvarez Pinera	Asesor Académico	300	\$0.00	\$0.00
Subtotal				\$3200.00

En la Tabla 42, se detalla la valoración económica de los servicios utilizados en el estudio comparativo.

Tabla 42.
Valoración Económica-Servicios

Servicio	Descripción	Precio unitario	(\$) Total
Internet	6 meses	\$25	\$150
Subtotal			\$150

La Tabla 43, muestra la valoración económica de los materiales de oficina utilizados en el presente trabajo.

Tabla 43.
Valoración Económica-Materiales de oficina

Nombre	Cantidad	Precio unitario	(\$) Total
Materiales de oficina	1	\$10	\$10
Subtotal			\$10

En la Tabla 44 y Tabla 45, se muestra los recursos de hardware y software utilizados en el estudio comparativo.

Tabla 44.
Valoración Económica- Recursos de Hardware

Nombre	cantidad	Precio unitario	(\$) Total
Laptop dell	1	\$12000	\$1200
Smartphone	1	\$300	\$300
Subtotal			\$1500

Tabla 45.
Valoración Económica- Recursos de Software

Nombre	Precio unitario	(\$) Total
Android Studio	\$0.00	\$0.00
Visual Code	\$0.00	\$0.00
Firebase	\$0.00	\$0.00
Figma	\$0.00	\$0.00
Zoom	\$0.00	\$0.00
Subtotal		\$0.00

En la Tabla 46, se puede observar el costo total del proyecto.

Tabla 46.
Costo total del proyecto

Recursos	Total
Talento Humano	\$3200.00
Servicios	\$150.00
Materiales de oficina	\$10.00
Recursos de Hardware	\$1500.00
Recursos de Software	\$0.00
Total	\$4860

8. Conclusiones

Al finalizar el presente Trabajo de Titulación se pudo cumplir de manera exitosa los objetivos propuestos inicialmente, y en este proceso de aprendizaje teórico y práctico se puede concluir lo siguiente:

- La Revisión Sistemática de Literatura (RSL), evidenció las metodologías de desarrollo de software más utilizadas durante la pandemia del covid-19 para la creación de aplicaciones móviles, obteniendo como resultado a Prototipado como la metodología de desarrollo más utilizada, seguida de la metodología en Cascada, Mobile-D, Metodología Híbrida (Scrum-XP), RUP y RAD, por ende, Prototipado fue la metodología seleccionada como objeto de estudio.
- De acuerdo a los artículos seleccionados en la Revisión Sistemática de Literatura, para la elección de una metodología de desarrollo de software para aplicaciones móviles, un equipo de desarrollo considera que esta garantice un producto de calidad, con pocos recursos y que se adapte a las necesidades del cliente debido a que pueden ser cambiantes.
- El proceso experimental desarrollado en el presente TT fue de gran importancia, debido a que se pudo aplicar la metodología Prototipado en un entorno turbulento bajo restricciones de la pandemia del covid-19, trabajando con sujetos reales aptos para el experimento donde se obtuvo resultados satisfactorios.
- El desarrollo de la aplicación de prueba fue un factor relevante durante el desarrollo del proceso experimental, debido a que se tomaron métricas de evaluación, desde la especificación de requerimientos hasta la entrega del producto final.
- El trabajo remoto ejecutado con el grupo de sujetos seleccionados, no afectó el proceso experimental donde se aplicó Prototipado, pese a que este tipo de metodología se trabaja de forma presencial y por tal motivo fue adaptada bajo las restricciones de la pandemia del covid-19.
- El estudio comparativo permitió determinar que Prototipado es una metodología de desarrollo de software que se adaptó al desarrollo de aplicaciones móviles, de acuerdo a los criterios de Usabilidad, Aplicabilidad, Procesos y Productos, y Capacidad que conformaron la matriz de evaluación, puesto que, de acuerdo a los resultados, esta metodología obtuvo una valoración del 91%, teniendo un grado de aceptación muy satisfactorio.

- En base a la RSL, el proceso experimental y la evaluación realizada en este TT, antes aplicar Prototipado, es fundamental realizar un análisis sobre el entorno de trabajo, capacidad y tamaño del equipo, el tiempo límite para el proyecto, y el tipo de aplicación o sistema a desarrollar, estos fueron factores clave que ayudaron a responder la pregunta de investigación.

9. Recomendaciones

Luego de realizar el presente trabajo de titulación se puede recomendar lo siguiente:

- Realizar una investigación exhaustiva sobre herramientas que faciliten la gestión y el desarrollo de aplicaciones móviles, ya que esto permitirá ahorrar tiempo y mitigar algunos riesgos suscitados durante la ejecución del proyecto, por el simple hecho de no tener claro con qué se puede trabajar de la mejor manera.
- Para determinar de mejor manera una metodología adecuada para el desarrollo de aplicaciones móviles, se recomienda trabajar con más de dos grupos de trabajo y desarrollar más tipos de aplicaciones, así se pueden tener mejores resultados y criterios más sólidos.
- Si se va a realizar una experimentación, así como se hizo en el presente Trabajo de Titulación, es decir seleccionando grupos de trabajo con personas reales, se recomienda hacer un análisis sobre la disponibilidad de cada grupo, ya que es muy importante que cada integrante tenga el tiempo, la responsabilidad y el compromiso de cumplir con que se le asigna, así evitando problemas y paros repentinos en el desarrollo del proyecto.
- Utilizar el presente trabajo de titulación, como fuente de información, debido a que contiene conocimientos obtenidos con base en la experiencia en el momento de gestionar equipos que trabajan con metodologías diferentes para el desarrollo una aplicación móvil.

Trabajos Futuros

Luego de haber concluido el presente TT se plantea los trabajos futuros, los cuales pueden ser considerados para una futura realización de los mismos:

- Realizar un estudio comparativo aplicado en un ámbito empresarial en un entorno donde se dediquen netamente al desarrollo de software, de esta manera se podrá obtener resultados más sólidos en una evaluación comparativa.
- Desarrollar una comparativa donde se evalúen más de dos metodologías de desarrollo de software para diferentes tecnologías, ya sean aplicaciones de escritorio, web o aplicaciones móviles.
- Realizar una investigación donde se comparen las metodologías de desarrollo de software aplicadas en entornos académicos y otras en entornos empresariales.

10. Bibliografía

- [1] E. Maida, «Metodologías de desarrollo de software,» Biblioteca digital de la universidad Católica de Argentina, Buenos Aires, 2021.
- [2] M. J. Pérez, «Guía comparativa de Metodologías Ágiles,» Universidad de Valladolid, Valladolid, 2012.
- [3] A. Gordillo, «Metodologia y metodo desde trabajo Social,» *Revista Tendecnai y Retos*, vol. 1, nº 12, pp. 119-135, 2007.
- [4] R. Aguilera, «Identidad y diferenciación entre Método y Metodología,» *Estudios Politicos*, vol. 9, nº 28, pp. 81-103, 2013.
- [5] O. Tinoco, «Criterios de seleccion de metodologías de desarrollo de software,» *Industrial Data*, vol. 13, nº 2, pp. 70-74, 2010.
- [6] R. Figueroa, «Metodologías tradicionales vs. Metodologías ágiles,» UTPL, Loja, 2008.
- [7] S. Amaro, «Metodologías Ágiles,» Universidad de Trujillo, Trujillo-Perú, 2007.
- [8] A. Alliance, «Manifiesto ágil,» 20 enero 2001. [En línea]. Available: <https://agilemanifesto.org/iso/es/manifesto.html>.
- [9] D. Cantone, *Implementacion y Debugging*, Mexico: Publicaciones CITEM, 2006.
- [10] ISO12207, «Systems and software engineering — Software life cycle processes,» 03 enero 2017. [En línea]. Available: <https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:12207:ed-1:v1:en>.
- [11] R. Campaña, «El proceso de desarrollo rápido de aplicaciones de software: Un aporte práctico en el Instituto Geográfico Militar,» El Dorado, Quito, 2015.
- [12] J. Martínez, «Desarrollo rápido de aplicaciones y puesta a punto,» *Revista Vinculos*, vol. 8, nº 1, pp. 1-8, 2011.
- [13] J. C. Rueda, «Aplicación de la metodología RUP para el desarrollo rápido de aplicaciones basados en el estándar J2EE,» Universidad San Carlos Guatemala, Ciudad de Guatemala, 2006.
- [14] J. Cervantes y M. d. C. Gómez, «Taxonomía de los modelos y metodologías de desarrollo de software más utilizados,» *Universidades*, vol. 1, nº 52, pp. 37-47, 2012.
- [15] D. Gutiérrez, «Métodos de Desarrollo de software,» Universidad de los Andes, Mérida, Venezuela, 2011.
- [16] M. Trigas, «Metodología SCRUM,» Universidad Oberta de Catalunya, Catalunya, España, 2012.

- [17] J. Palacio, «Scrum Manager I,» Lubaris, 2015.
- [18] M. Palacio, «Scrum Master Temario Troncal 1,» Iubaris Info 5 Media SL, 2021.
- [19] Y. Amaya, «Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles,» Universidad pedagógica y tecnológica de Colombia, Tunja, Colombia, 2013.
- [20] StatCounter, «Statcounter GlobalStats,» 30 diciembre 2021. [En línea]. Available: <https://gs.statcounter.com/os-market-share/mobile-tablet/worldwide/#monthly-202001-202012-bar>.
- [21] G. Pedrozo, «Sistemas operativos,» Universidad Nacional del Nordeste, Corrientes, Argentina, 2012.
- [22] W. Zambrano, «Estudio Comparativo de Metodologías de Desarrollo Ágil en Base al Desarrollo de una Aplicación Móvil,» UCE, Quito, 2017.
- [23] A. F. Julio Cameron, «Metodología de desarrollo ágil para sistemas móviles: Introducción al desarrollo con Android y el iPhone,» UPM, Madrid, España, 2009.
- [24] E. Méndez, «Modelo de Evaluación de Metodologías para el desarrollo de software,» Universidad Católica Andrés Bello, Caracas, Venezuela, 2006.
- [25] G. López, «Estudio comparativo de metodologías tradicionales y ágiles para proyectos de Desarrollo de Software,» Escuela de Ingenierías Industriales, Valladolid, 2018.
- [26] M. Genero, J. Cruz y M. Piattini, Métodos de investigación en ingeniería del software, Madrid, España: RA-MA, S.A, 2014.
- [27] C. Tebes, P. Becker, D. Peppino y L. Olsina, «Especificación del Modelo de Proceso para una Revisión Sistemática de Literatura,» ResearchGate, La Pampa, Argentina, 2019.
- [28] J. G. Navarro, «Estudio Comparativo de Metodologías, Herramientas y Wiki de soporte para la gestión de proyectos de desarrollo de software,» Universidad de Catalunya, Catalunya, 2018.
- [29] X. Quiñonez y J. Casierra, «Análisis comparativo de metodologías ágiles de desarrollo de software: una revisión bibliográfica,» Pontificia Universidad Católica Sede Esmeraldas, Esmeraldas, 2019.
- [30] A. V. R. Campuzano, «Aplicaciones digitales de delivery: Incidencia en el consumo de comidas rápidas durante el primer semestre del año 2020 en Guayaquil,» Universidad Politécnica Salesiana, Guayaquil, 2021.
- [31] J. L. Becerra, «CIO MÉXICO,» 24 abril 2020. [En línea]. Available: <https://cio.com.mx/que-apps-han-tenido-mas-demanda-en-esta-cuarentena/>.
- [32] M. Panier, «PICKASO,» 16 abril 2020. [En línea]. Available:

<https://pickaso.com/2020/covid-19-apps-juegos-mas-descargados-google-play>.

- [33] R. Peco, «La Vanguardia,» 21 abril 2020. [En línea]. Available: <https://www.lavanguardia.com/tecnologia/20200421/48647266319/aplicaciones-apps-herramientas-servicios-personas-viven-solas-grupos-de-riesgo-covid-19-pandemia-coronavirus.html>.
- [34] F. García y A. García, «Fundamentos de la vista de casos de uso,» universidad de Salamanca, Salamanca, 2018.
- [35] R. Espinosa, «Análisis comparativo para la evaluación de frameworks usados en el desarrollo de aplicaciones web,» Universidad Nacional de Loja, Loja, 2022.
- [36] M. A. Sobrino, «Historias de Usuario,» Scrum Manager, 2020.

11. Anexos

Anexo 1: Revisión Sistemática de Literatura

Revisión Sistemática de Literatura: Metodologías de desarrollo de software usadas durante la pandemia del covid19 para el desarrollo de aplicaciones móviles.

Universidad Nacional de Loja
Carrera de Ingeniería en Sistemas

Andy Aron Camacho Herrera
andy.camacho@unl.edu.ec

Resumen

La presente Revisión Sistemática de Literatura tiene como objetivo identificar las metodologías de desarrollo de software usadas durante la pandemia por el covid-19 orientadas al desarrollo de aplicaciones móviles. Para llevar a cabo este objetivo se ha seguido una metodología basada en la de Barbara Kitchenham. Se ha hecho uso de 4 bases de datos científicas como Scopus, Web of Science, IEEE Xplore y ACM Library las cuales aumenta la confiabilidad, evidencia y calidad de la información.

Se aplican criterios de inclusión, exclusión y 2 análisis de calidad, quedándonos con 24 artículos para analizar los cuales permitieron responder a las preguntas de investigación planteadas en esta revisión.

Como conclusión se obtiene que las metodologías utilizadas en el desarrollo de software durante la pandemia del covid-19 para crear aplicaciones móviles son: Scrum con un 34%, Prototipado con 29%, Cascada con 13%, Metodologías híbridas (Scrum-XP) 8%, Mobile-D con 8%, RUP con 4%, y RAD con 4%.

Abstract

This Systematic Literature Review aims to identify the software development methodologies used during the covid-19 pandemic oriented to the development of mobile applications. To achieve this objective, a methodology based on Barbara Kitchenham's methodology has been followed. Use was made of 4 scientific databases such as Scopus,

Web of Science, IEEE Xplore and ACM Library, which increase the reliability, evidence and quality of the information. Inclusion, exclusion and 2 quality analysis criteria were applied, leaving us with 24 articles to analyze, which allowed us to answer the research questions posed in this review.

As a conclusion it is obtained that the methodologies used in software development during the covid-19 pandemic to create mobile applications are: Scrum with 34%, Prototyping with 29%, Waterfall with 13%, Hybrid methodologies (Scrum-XP) 8%, Mobile-D with 8%, RUP with 4%, and RAD with 4%.

Keywords: agile methodologies, covid-19, development methodology, mobile application, Software development process.

1.Introducción

El proceso de desarrollo de software debe tener como propósito fundamental una producción de calidad que reúna los requisitos y satisfaga las necesidades del cliente y/o usuario al que va dirigido. Dicho proceso se denomina metodología; se constituye de una combinación de métodos existentes (desarrollo en espiral, incremental, cascada, ciclo de vida, reutilización y desarrollo evolutivo), que en ingeniería de software dan origen a las etapas o fases de producción [1].

Actualmente con esta pandemia, universidades, instituciones pública o privadas, y empresas orientadas al desarrollo de software han tenido que adaptarse a nuevas modalidades para poder mantener el trabajo que han llevado haciendo por años, por lo que han tenido que adaptarse a nuevas o ya conocidas metodologías de desarrollo de software para poder llevar a cabo sus proyectos y productos demandados por los usuarios ajustándose a las nuevas normas [2].

El tipo de aplicaciones más demandadas en esta pandemia fueron las aplicaciones web y móviles, debido a que las personas tenían que permanecer en sus casas respetando el horario de toque de queda para evitar la propagación del virus, y este tipo de aplicaciones les brindaban diferentes servicios a los usuarios, entre estos se encontraban el streaming, servicio de comida a domicilio, servicio de música, videojuegos entre otros [3].

La Revisión Sistemática de Literatura se centra en identificar las metodologías de desarrollo de software usadas durante la pandemia del covid-19 para crear aplicaciones

móviles, con el fin de determinar que metodologías optaron las universidades o empresas para desarrollar sus productos y cumplir su demanda bajo esta nueva modalidad.

2. Metodología

La metodología aplicada en esta Revisión Sistemática de Literatura fue la propuesta por Barbara Kitchenham lo cual implica planificar, ejecutar y documentar la revisión, estas fases son adaptadas y detalladas en el presente proyecto.

En la Figura 1, se puede observar todo el proceso ejecutado para esta revisión sistemática de literatura, donde se describe cada uno de los pasos seguidos de acuerdo a la metodología anteriormente mencionada.

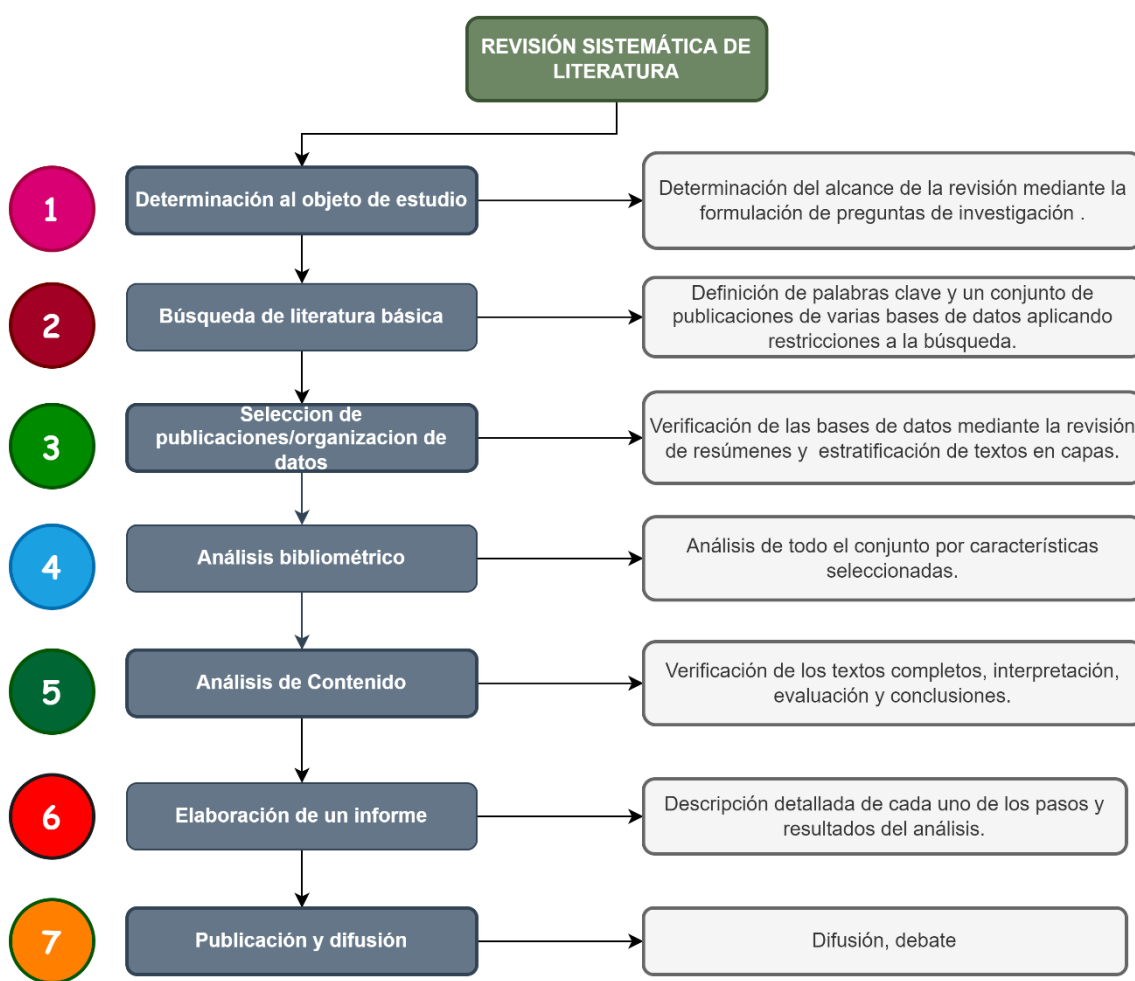


FIGURA I
Proceso de la Revisión Sistemática de Literatura

2.1 Preguntas de investigación

En la TABLA I se plantean las preguntas de investigación de acuerdo con el objetivo de la RSL.

TABLA I
PREGUNTAS DE INVESTIGACIÓN

ID	Preguntas de Investigación
P1	¿Qué metodologías de desarrollo de software se usaron durante la pandemia del covid-19 para crear aplicaciones móviles?
P2	¿Qué criterios mencionan los autores para la elección de una metodología de desarrollo para aplicaciones móviles?
P3	¿Para qué tipo de Sistema Operativo fueron desarrolladas las aplicaciones móviles de acuerdo a los estudios seleccionados?

2.2 Estrategia de búsqueda

a. Definir palabras claves

Para la realización de la RSL se utilizó la siguiente lista de palabras claves, ya que estas permitieron a realizar una búsqueda de información más exhaustiva y eficiente:

- Software developement process.
- University
- Companies
- Agile methodology
- development methodology.
- mobile.
- mobile application.
- coronavirus

b. Cadenas de búsqueda

Se realizaron las consultas mediante operadores lógicos AND y OR generando las cadenas de búsqueda que se muestran en la TABLA II.

TABLA II
CADENAS DE BÚSQUEDA

ID	Cadena de Búsqueda
CB_1	(university OR institute OR companies) AND ("agile methodologies" OR methods) AND ("mobile devices" OR "mobile applications")
CB_2	(university OR institute OR companies) AND ("agile methodologies" OR methods) AND ("mobile devices" OR "mobile applications") AND (Coronavirus)
CB_3	("software development process" OR "agile methodology" OR "development methodology") AND ("Software Development") AND ("mobile" OR "mobile application")

c. Bases de datos

Se utilizaron cuatro bases de datos científicas, las cuales se describen en la TABLA III.

TABLA III
BASE DE DATOS

Base de Datos	URL
WoS	http://www.isiknowledge.com
Scopus	http://scopus.com/
IEEE	https://ieeexplore.ieee.org/
ACM	https://dl.acm.org/

2.3 Ejecutar las cadenas de búsqueda

Las cadenas iniciales (TABLA II) fueron ajustadas para cada base de datos, ya que la sintaxis aceptada por cada una de ellas tiene una variación de aceptación (ver apéndice 1). Además, se utilizó la herramienta parsifal para gestionar las referencias bibliográficas obtenidas de las cadenas de búsqueda, se ha logrado identificar un total de 308 trabajos

(ver apéndice 2). Las cadenas aceptadas en cada base de datos y los resultados proporcionados por las mismas se indican en la TABLA IV.

TABLA IV
EJECUCIÓN DE CADENAS DE BÚSQUEDA

Base de Datos	Cadena de Búsqueda	Número de Resultados
Scopus	("university" OR "companies") AND ("software development process" OR "agile methodology" OR "development methodologies") AND ("Software Development") AND ("mobile" OR "mobile application") AND (LIMITTO (OA , "all")) AND (LIMITTO (PUBYEAR , 2021) OR LIMITTO (PUBYEAR , 2020))	154
IEEE Xplorer	((("All Metadata":university OR companies) AND "All Metadata":mobile o mobile aplicacion) AND "All Metadata":"software development process" OR "agile methodology" OR "development methodolody")	50
WoS	ALL=(university OR companies) AND ALL=("software development process" OR "agile methodology" OR "development methodology") AND ALL=("mobile" OR "mobile application")	15
ACM	AllField:(university OR companies) AND AllField:("software development process" OR "agile methodology" OR "development methodology") AND AllField:(mobile OR "mobile application")	89
TOTAL		308

2.4 Criterios de selección

a. Criterios de Inclusión

Los criterios de inclusión que se utilizaron se describen en la TABLA V.

TABLA V
CRITERIOS DE INCLUSIÓN

Criterio	Descripción
Contenido	Debe contener al menos en una de las cadenas de búsqueda en su título y una o algunas en su resumen.
Fecha de publicación	Estudios publicados desde el año 2020-2021
Motores de búsqueda	WoS, Scopus, IEEE Xplore, ACM.
Idioma	Se consideran artículos estrictamente en inglés, tomando en cuenta un porcentaje bajo en español.
Tipos de estudios	Artículos de revistas y artículos de conferencias.

b. Criterios de Exclusión

No serán considerados los estudios, si no cuentan con información clave que aporten a responder las preguntas de investigación y que no cumplan con los criterios de inclusión, que se muestran en la TABLA V.

2.5 Proceso de selección

a. Análisis 1: Selección de estudios primarios.

En esta etapa se analizó únicamente el título y el resumen de los artículos relacionados en donde se decidió si el artículo es incluido o excluido, de acuerdo a los criterios de inclusión y exclusión definidos (ver sección 2.4). Para su respectivo análisis se utilizó la herramienta Parsifal (ver apéndice 3) y el resultado se muestra en la TABLA VI.

TABLA VI
ANÁLISIS 1: Selección de estudios primarios

Base de Datos	Artículos Aprobados
Scopus	24
IEEE Xplore	8
WoS	5
ACM	10
Total:	47

b. Análisis De Lectura crítica y control de calidad

En esta fase se aplicó a los resultados obtenidos del análisis anterior (ver TABLA VI) el cuestionario de calidad, que constan de 4 preguntas (ver apéndice 4). Los artículos que superen el punto de corte del 75% del cuestionario de calidad aplicado, serán los que se usarán para responder las preguntas de la RSL, además, para el análisis se lo realizó mediante la herramienta Parsifal (ver apéndice 5). El resultado obtenido se muestra en la TABLA VII.

TABLA VII
ANÁLISIS DE LECTURA CRÍTICA Y CONTROL DE CALIDAD

Base de Datos	Artículos Aprobados
Scopus	15
IEEE Xplore	1
WoS	5
ACM	3
Total:	24

c. Síntesis del proceso de selección de los estudios.

En la Figura 2, se puede observar el proceso de selección de los estudios primarios seleccionados de manera sintetizada, eso con el fin de que el lector entienda de una mjero manera este proceso.

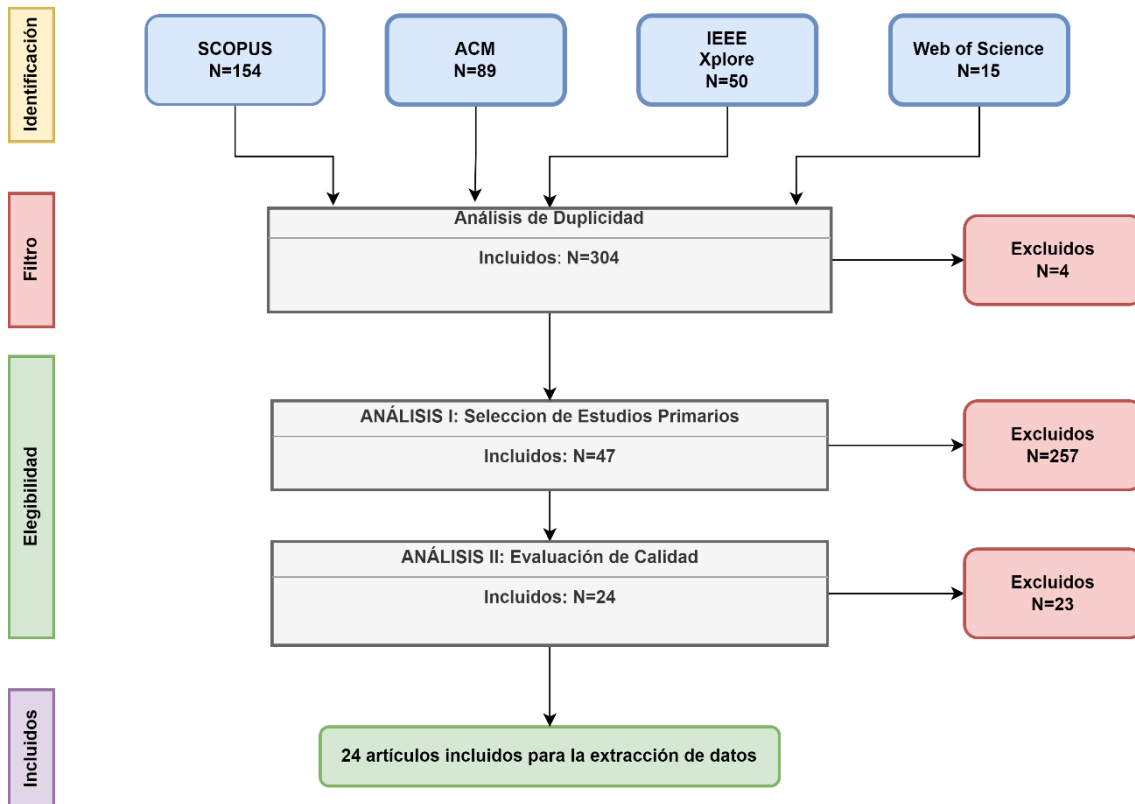


FIGURA II
Proceso de la Revisión Sistemática de Literatura

Luego con ayuda de la herramienta parsifal, se procedió a extraer la información importante de los estudios relacionados con respecto a las preguntas de investigación. Estos se presentan en la Tabla IX:

TABLA IX
Extracción de Datos de los Artículos Seleccionados

Artículo	Base de Datos Científica	Fecha de publicación	Tipo de Metodología	Nombre de la metodología	Criterios de selección	Sistema operativo
Automated Test Input Generation via Model Inference Based on User Story and Acceptance Criteria for Mobile Application Development	Web of Science	2020	Ágil	Scrum	calidad del producto, ciclos cortos, desarrollo ágil	Multiplataforma
An Agile-Based Integrated Framework for Mobile Application Development Considering Ilities	Web of Science	2020	Ágil	Scrum	capacidad de agilidad, satisfacción del cliente	Android
EmERGE mHealth Platform: Implementation and Technical Evaluation of aDigital Supported Pathway of Care for Medically Stable HIV	Web of Science	2021	Híbrida	Hybrid(Scrum-XP)	desarrollo incremental, calidad del producto, adaptable a cambios, proyectos pequeños/medianos	Multiplataforma
Realtime Notifications on Visitor Tracking Systems Using Android and Arduino	SCOPUS	2020	Tradicional	Cascada	calidad del producto	Android
Thailand medical mobile application for patients triage base on criteria based dispatch protocol	SCOPUS	2020		Cascada	satisfacción del cliente	Multiplataforma

CardioResyncApp: A mobile phone application for recording research data in Cardiology [CardioResyncApp: Un aplicativo móvil para recolectar datos de investigación en Cardiología]	SCOPUS	2020	Ágil	Scrum	ciclos cortos, desarrollo incremental, colaboración, capacidad de agilidad	Multiplataforma
The Necessity of Interdisciplinary Software Development for Building Viable Research Platforms: Case Study in Automated Drug Delivery in Diabetes	ACM	2020	Tradicional	Prototipado	desarrollo iterativo, calidad del producto, proyectos grandes	Android
Developing a Design Phase for a Mentoring Mobile App	ACM	2020	Tradicional	Prototipado	desarrollo iterativo	Multiplataforma
Close2U: An App for Monitoring Cancer Patients with Enriched Information from Interaction Patterns	Web of Science	2020	Tradicional	Prototipado	desarrollo iterativo e incremental	Android
Using goal-directed design to create a mobile health app to improve patient compliance with hypertension self-management: Development and deployment	SCOPUS	2020	Tradicional	Prototipado	desarrollo incremental, producto funcional	Multiplataforma
Systems dynamics and serious video games in an	SCOPUS	2020	Tradicional	RUP	documentación adecuada, calidad	Android

appropriation strategy of a decision support system for small livestock farmers					del producto, desarrollo iterativo, proyectos grandes	
Multiplatform system to improve tourist services in the city of trujillo [Sistema Multiplataforma para mejorar los Servicios Turísticos en la ciudad de Trujillo]	SCOPUS	2020	Híbrida	Hybrid(Scrum-XP)	colaboración, capacidad de agilidad	Multiplataforma
Mobile application to improve tourist flow in trujillo [Aplicación Móvil para Mejorar el Flujo Turístico en Trujillo]	SCOPUS	2020	Ágil	Mobile-D	capacidad de agilidad, calidad del producto	Android
Investigating agile adoption in saudi arabian mobile application development	SCOPUS	2020	Ágil	Mobile-D	proyectos pequeños/medianos, desarrollo ágil	Multiplataforma
Doodle2App: Native app code by freehand UI sketching	SCOPUS	2020	Tradicional	Prototipado	proyectos grandes	Android
Application Information System Smart Parking Based on Smartphone	SCOPUS	2020	Tradicional	Cascada	calidad del producto	iOS
A Mobile Serious Game About the Pandemic (COVID-19 - Did You Know?): Design and Evaluation Study	SCOPUS	2020	Ágil	Scrum	desarrollo iterativo, desarrollo incremental	Multiplataforma

A software to prevent delirium in hospitalised older adults: Development and feasibility assessment	SCOPUS	2020	Ágil	Scrum	colaboración, roles intercambiables, adaptable a cambios	Android
A mobile health approach for improving outcomes in suicide prevention (SafePlan)	SCOPUS	2020	Tradicional	Prototipado	proyectos grandes	Multiplataforma
A comparative analysis of RAD and agile technique for management of computing graduation projects	SCOPUS	2020	Tradicional	RAD	documentación adecuada	Android
Design of a mobile application for the school enrollment process in order to prevent Covid-19	SCOPUS	2020	Ágil	Scrum	adaptable a cambios, satisfacción del cliente	Android
Troubadour: A Gamified e-Learning Platform for Ear Training	Web of Science	2020	Tradicional	Prototipado	satisfacción del cliente	Android
Agile Methodology Advantages when delivering Internet of Things projects	IEEE	2020	Ágil	Scrum	desarrollo incremental, desarrollo iterativo, colaboración en equipo, roles intercambiables	Multiplataforma
Scrum to Support Application Development Project for Online Learning	ACM	2020	Ágil	Scrum	adaptable a cambios, satisfacción del cliente	Android

3.Resultados

Después de haber realizado los análisis correspondientes en sus dos fases, se obtuvieron un total de 19 artículos, que es la evidencia empírica, para responder las preguntas de investigación que se plantearon en la RSL:

Pregunta 1: ¿Qué metodologías de desarrollo de software se usaron durante la pandemia del covid-19 para crear aplicaciones móviles?

Las metodologías en el desarrollo de software han estado muy presentes en la creación de aplicaciones móviles independientemente del tipo dispositivo y aún más durante la pandemia del covid-19, la cual tuvo un gran impacto en el desarrollo de software en general.

Estas metodologías ayudan a que los proyectos tengan éxito, es decir se mantengan en el transcurso del tiempo por medio de un funcionamiento estable y actualizaciones disponibles, no solo durante esta etapa de confinamiento, sino que se sigan mejorando las funcionalidades y se mantenga estable en el mercado aun después cuando se termine la pandemia.

En los papers analizados se han podido extraer el tipo de metodologías de desarrollo usadas en la creación de aplicaciones móviles durante la pandemia del covid-19.

Los resultados obtenidos se muestran en el siguiente FIGURA III.

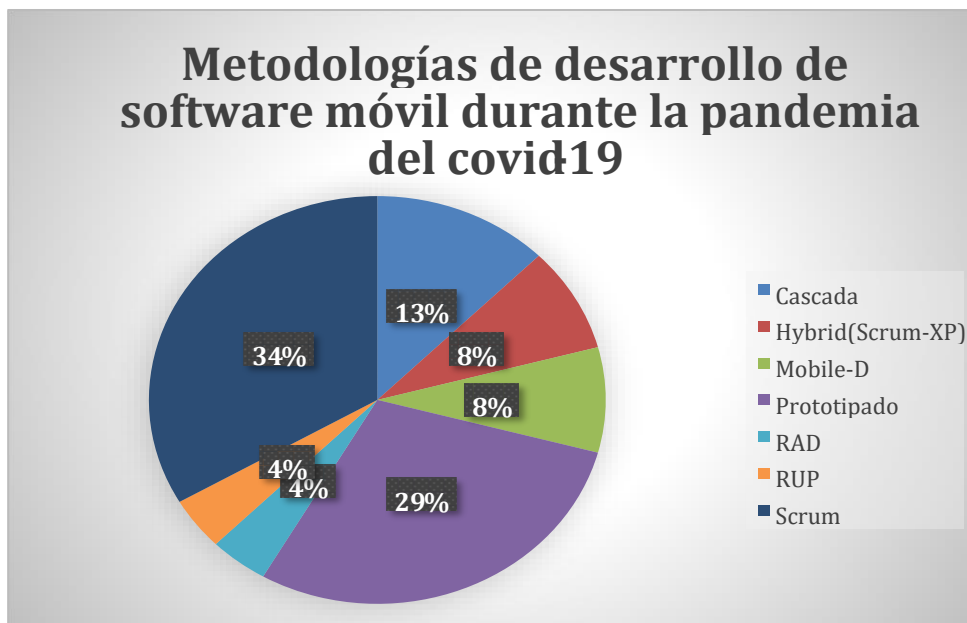


FIGURA III
Metodologías de desarrollo de software usadas durante la pandemia del Covid-19

De los 24 papers analizados se obtiene que las metodologías utilizadas en el desarrollo de software durante la pandemia del covid-19 para crear aplicaciones móviles son: Scrum con un 34%, Prototipado con 29%, Cascada con 13%, Metodologías híbridas (Scrum-XP) 8%, MobileD con 8%, RUP con 4%, y RAD con 4%.

En la siguiente TABLA X, se indica el nombre y la frecuencia de las metodologías utilizadas para el desarrollo de software móvil.

TABLA X
Frecuencia de Metodologías de desarrollo de software para aplicaciones móviles

<i>Tipo de metodología</i>	<i>Nombre</i>	<i>Frecuencia</i>
TRADICIONALES	Prototipado	7
	Cascada	3
	RUP	1
	RAD	1
ÁGILES	Scrum	8
	Mobile-D	2
HÍBRIDAS	Scrum/XP	2

Pregunta 2: ¿Qué criterios mencionan los autores para la elección de una metodología de desarrollo para aplicaciones móviles?

Como se puede observar en la FIGURA IV, se ha obtenido un total de 12 criterios que los autores han tomado en cuenta para la elección de una metodología en el desarrollo de su proyecto.

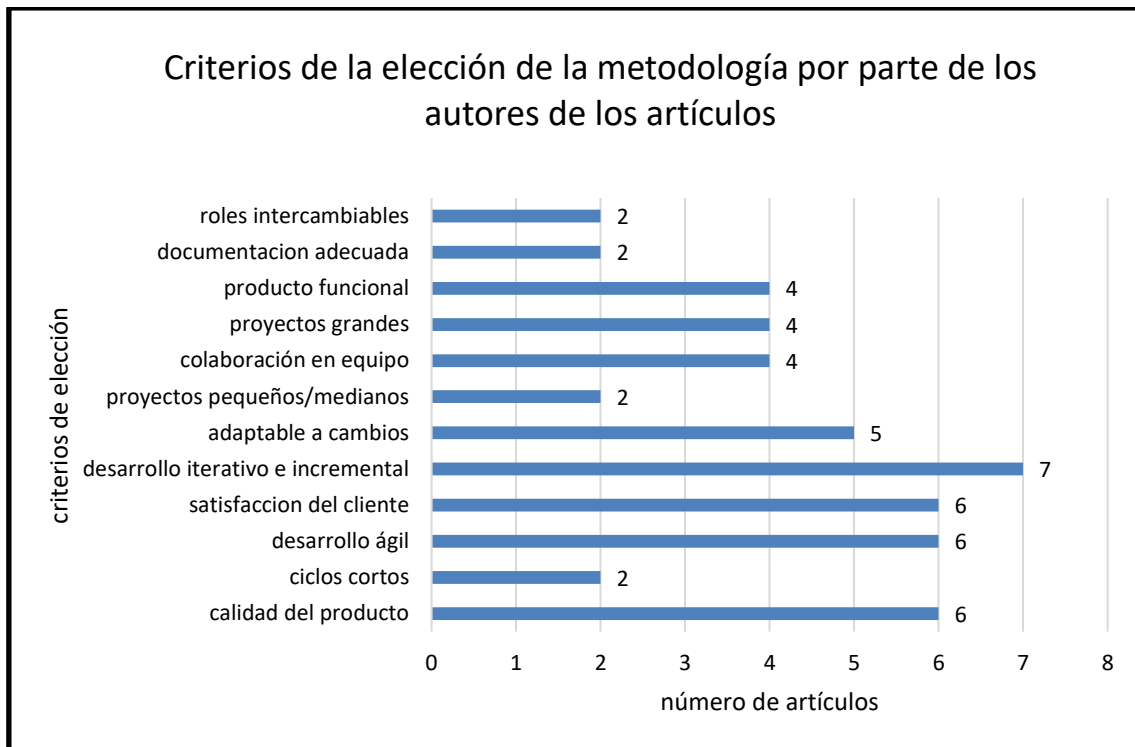


FIGURA IV
Criterios de selección de las metodologías de desarrollo móvil

Los 12 criterios que se obtuvo son:

- Roles intercambiables
- Documentación adecuada
- Producto funcional
- Proyectos grandes
- Colaboración en equipo
- Proyectos pequeños/medianos
- Adaptable a cambios
- Desarrollo iterativo e incremental
- Satisfacción del cliente
- Desarrollo ágil
- Ciclos cortos
- Calidad del producto

Pregunta 3: ¿Para qué tipo de Sistema Operativo fueron desarrolladas las aplicaciones móviles de acuerdo a los estudios seleccionados?

En la FIGURA V, se puede observar que en la mayoría de artículos desarrollan apps para dispositivos Android, y también Multiplataforma.

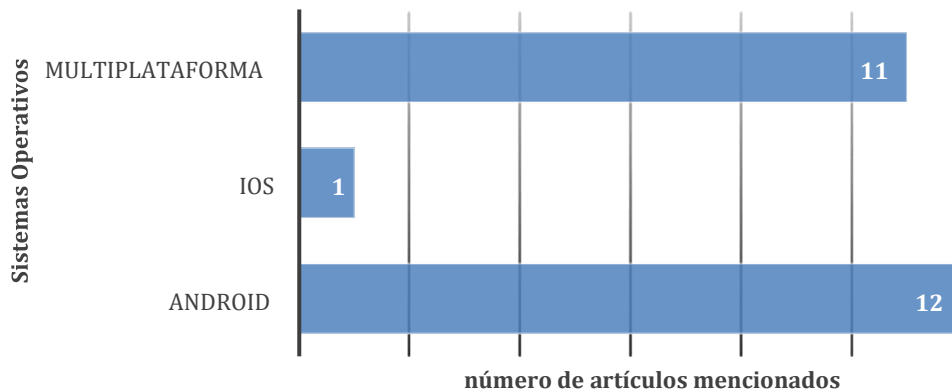


FIGURA V
Sistemas operativos para aplicaciones móviles

Como se pueden observar en la Figura 3, Android lidera el desarrollo de aplicaciones móviles con un 50% de los papers analizados. Por otra parte, también consideran desarrollar aplicaciones multiplataforma que sean compatibles con los dos sistemas operativos que lideran a nivel mundial, los cuales son Android y iOS.

Estos resultados son importantes, debido a que ayudan a formar un escenario real el cual se va a comprobar con un entorno de experimentación donde se evaluarán metodologías de desarrollo móvil por medio de la creación de una aplicación para dispositivos móviles.

4. Discusión

Según el artículo “The Necessity of Interdisciplinary Software Development for Building Viable Research Platforms: Case Study in Automated Drug Delivery in Diabetes” [4], utiliza “Prototype Method” para desarrollar una aplicación móvil destinada al control de la salud de pacientes que padecen diabetes, así mismo el artículo “Close2U: An App for Monitoring Cancer Patients with Enriched Information from Interaction Patterns” [5], indica el desarrollo de una aplicación destinada al monitoreo de pacientes con cáncer, el cual utiliza “Prototype Method”. Otro artículo que hace uso es la misma metodología es “Developing a Design Phase for a Mentoring Mobile App” [6], el cual crean un prototipo de una aplicación móvil para tutores y estudiantes de la universidad de Salamanca, España. Otro artículo que hace referencia al uso de la metodología antes mencionada es “Troubadour: A Gamified e-Learning Platform for Ear Training” [7], en este artículo desarrollan una aplicación destinada para docentes y estudiantes de la Universidad de Ljubljana, Slovenia, para apoyar las clases teóricomusical de los antes mencionados. “Using goal-directed design to create a mobile health app to improve patient compliance with hypertension self-management: Development and deployment” [8], en este artículo desarrollan un prototipo, el cual se trata de una aplicación móvil para pacientes que sufren hipertensión, donde por medio de esta app móvil, controla el estado de salud del paciente otro artículo donde usan la misma metodología es “Doodle2App: Native app code by freehand UI sketching” [9], la cual crean una aplicación híbrida (web-móvil), donde los usuarios finales de la aplicación pueden crear bocetos o dibujos, con el fin de sustituir el papel. Por último, tenemos el artículo “A mobile health approach for improving outcomes in suicide prevention (SafePlan)” [10], el objetivo en este paper es desarrollar una aplicación móvil que facilite el acceso a los usuarios de los servicios que necesitan ayuda en su salud mental y brindarle mayor seguridad.

Existen 8 artículos donde utilizan Scrum para el desarrollo de sus aplicaciones móviles, entre ellos están:

- “Automated Test Input Generation via Model Inference Based on User Story and Acceptance Criteria for Mobile Application Development” [11] donde indican algunos criterios de aceptación para usar este marco de trabajo para el desarrollo

de aplicaciones móviles.

- “Agile Methodology Advantages when delivering Internet of Things projects” [12], este artículo tiene un enfoque en dar algunas recomendaciones y ventajas de utilizar Scrum para desarrollar aplicaciones móviles orientas a IoT.
- “An Agile-Based Integrated Framework for Mobile Application Development Considering Ilities” [13], es un estudio que analiza los desarrolladores que utilizan Scrum para el desarrollo de aplicaciones móviles.
- “CardioResyncApp: A mobile phone application for recording research data in Cardiology [CardioResyncApp: Un aplicativo móvil para recolectar datos de investigación en Cardiología]” [14], en este artículo utilizan Scrum para desarrollar una aplicación móvil que registre datos de investigación del área de cardiología.
- “A Mobile Serious Game About the Pandemic (COVID-19 - Did You Know?): Design and Evaluation Study” [15], este artículo desarrollan un juego serio donde proporcionan información sobre la prevención del Covid-19 y el cuidado personal surante la pandemia. Este artículo proporcionó información muy relevante sobre el tema de estudio debido a que habla sobre la pandemia, y como este equipo trabajó para desarrollar esta aplicación, lo cual sirvió para sacar los criterios de selección de la metodología.
- “A software to prevent delirium in hospitalised older adults: Development and feasibility assessment” [16], proporciona información sobre del desarrollo de una aplicación dirigida a adultos mayores hospitalizados con el fin de que puedan acceder más fácilmente a intervenciones médicas y a fármacos. Esta aplicación fue desarrollada para un hospital donde coordinaron todo el proceso de trabajo mediante la metodología Scrum, dándoles buenos resultados y con un producto funcional que en este caso era la aplicación móvil.
- “Design of a mobile application for the school enrollment process in order to prevent Covid-19” [17], en este artículo nos indica el proceso y la aplicación de la metodología Scrum para desarrollar una aplicación móvil cuyo objetivo es facilitar a los padres de familia la inscripción de sus hijos a su establecimiento

estudiantil en Perú, pues menciona que esta metodología se adapta correctamente a los cambios que se está viviendo por la pandemia. En esta aplicación el alumno podrá validar sus datos personales ya registrados, su historial escolar, sus notas, asistencia, cursos, horarios y pagos de la matrícula; dando buenos resultados y se adaptado a las necesidades.

- “Scrum to Support Application Development Project for Online Learning” [18], en este artículo muestran el proceso de desarrollo de una aplicación móvil para estudiantes que ofrecen servicios de enfermería, aplicando la metodología ágil Scrum. Indican que esta metodología se adapta a sus necesidades y al ambiente cambiante que ha generado la pandemia. También indican que los requisitos se han ido modificando deliberadamente, lo cual Scrum se adapta fácilmente a esto, y ha dado buenos resultados, tanto en gestión como en el desarrollo del producto final.

Por otra parte, existen dos artículos que han aplicado una metodología híbrida la cual combina Scrum y XP, estos artículos son:

- “EmERGE mHealth Platform: Implementation and Technical Evaluation of a Digital Supported Pathway of Care for Medically Stable HIV” [19], este artículo muestra el Desarrollo de una aplicación móvil dirigida para personas padeciente de VIH. Esta plataforma fue construida aplicando una metodología híbrida la cual es el marco de trabajo Scrum y XP, la cual aprovechan las ventajas de la misma para desarrollar esta app que les ayude tanto a los médicos como a los pacientes en el control de esta enfermedad. Esta app fue implementada en 5 países europeos, utilizada por 2252 pacientes y más de 20 médicos, dando buenos resultados y teniendo muy buena aceptabilidad por ambas partes participante en este proyecto.
- “Multiplatform system to improve tourist services in the city of trujillo [Sistema Multiplataforma para mejorar los Servicios Turísticos en la ciudad de Trujillo]” [20], muestra el desarrollo de una aplicación móvil y un sistema web destinada a servicios turísticos de la ciudad de Trujillo en Perú, cuyo fin es brindar información sobre los restaurantes, hoteles, agencias de turismo, sitios nocturnos, etc. Para desarrollar este sistema aplicaron la metodología XP junto con el marco de trabajo Scrum, los cuales les dio buenos resultados y obtuvieron un producto

funcional.

Algo interesante es que también existen artículos que han aplicado métodos tradicionales como la metodología en cascada, estos artículos son:

- “Realtime Notifications on Visitor Tracking Systems Using Android and Arduino” [21], aquí construyen un Sistema de seguridad para visitantes de una zona residencial donde por medio de una aplicación móvil notificará al usuario la presencia de un invasor a su vivienda si es este el caso. Utilizan la metodología en cascada la cual crean una aplicación que combina Android con Arduino, teniendo bien claro los requerimientos, dándoles buenos resultados en la construcción de este prototipo.
- “Thailand medical mobile application for patients triage base on criteria based dispatch protocol” [22], en este artículo muestra el proceso de desarrollo de una aplicación móvil para paciente prehospitalarios con el fin de agilizar el estado de emergencia de los hospitales en Tailandia. Esta aplicación fue desarrollada para Android como para iOS, siguiendo una metodología en cascada la cual les dio buenos resultados y aceptación de la app.
- “Application Information System Smart Parking Based on Smartphone” [23], aquí utilizan el método en cascada para desarrollar una aplicación para dispositivos Android que permite ver la disponibilidad de aparcamientos disponibles en un centro comercial. Esta aplicación notifica información sobre estacionamientos disponibles, reservas de estacionamiento, y pagos de estacionamiento.
- “Systems dynamics and serious video games in an appropriation strategy of a decision support system for small livestock farmers” [24], en este artículo desarrollan un juego serio para ganaderos. Esta aplicación simula una granja real donde los ganaderos podrán interactuar en esta, conociendo sus procesos, las herramientas y todo lo que tiene que ver con el mantenimiento de una granja. Aquí utilizan la metodología RUP, la cual implementan sus fases y llevan a cabo el desarrollo de la aplicación.
- “A comparative analysis of RAD and agile technique for management of

computing graduation projects” [25], en este artículo realiza una comparación entre la metodología RAD y una técnica ágil para gestionar proyectos de graduación. Esta comparativa le realizan mediante una aplicación en Android llamada “Life Organizer”, la cual sirvió como escenario de experimentación para la comparativa de estas dos metodologías. Como resultado dio que la metodología RAD destacó en un 13% mejor en proporcionar documentación con respecto a la técnica ágil aplicada.

En otros dos artículos en cambio aplican otra metodología ágil para aplicaciones móviles la cual es conocida como mobile-D, estos artículos son:

- “Mobile application to improve tourist flow in trujillo [Aplicación Móvil para Mejorar el Flujo Turístico en Trujillo]” [26] , en este artículo indican el proceso en el cual el desarrollo una aplicación móvil para mejorar el flujo turístico de Trujillo, Perú. Aquí aplican la metodología mobile-D cuyas fases son: exploración, inicialización, producción, estabilización, y pruebas del sistema. La implementación de esta metodología les dio resultados positivos, en el cual aumentaron en un 43,6% en satisfacción a los clientes finales, en este caso los turistas, los cuales por medio de la aplicación pudieron obtener información relevante respecto a los lugares turísticos de la ciudad.
- “Investigating agile adoption in saudi arabian mobile application development” [27], en este paper realizan una investigación sobre los beneficios de aplicar metodologías ágiles en la industria del software en Arabia Saudita, indicando que para el desarrollo de ciertas aplicaciones como los son las apps móviles, una de las mejores metodologías para este tipo de software, es Mobile-D, debido a que dan excelentes resultados y una muy buena satisfacción del cliente, y eso es en lo que más se enfoca una empresa encargada de vender software de calidad.

5. Conclusiones

Al finalizar la RSL se presenta las siguientes conclusiones:

- La Revisión de literatura Sistemática permitió identificar las metodologías aplicadas durante la pandemia del covid-19 para desarrollar aplicaciones móviles, donde se obtuvo que los siguientes resultados: Scrum con un 34%, Prototipado con 29%, Cascada con 13%, Metodologías híbridas (Scrum-XP) 8%, Mobile-D con 8%, RUP con 4%, y RAD con 4%, lo que nos quiere decir que la metodología más usada es Scrum.
- Las metodologías ágiles se encuentran muy presentes en el desarrollo de software móvil debido a que superan en un gran porcentaje de usabilidad a las metodologías tradicionales, pero esto no quiere decir que una sea mejor que la otra. Algunos autores recomiendan elegir la metodología de acuerdo al proyecto que se quiera realizar y al número de miembros con los que cuenta el equipo.
- Mas de 50% de los papers analizados toman desarrollan sus aplicaciones móviles para dispositivos Android, esto quiere decir que este sistema operativo está muy vigente y muy utilizado por la población. Sin embargo, un gran porcentaje también toman en cuenta en crear aplicaciones multiplataforma que sean compatibles con Android y con iOS.

Referencias

- [1] E. García, «Metodología para el desarrollo de software multimedia educativo MEDESME,» *CPU-e*, vol. 2, n° 23, p. 12, 2016.
- [2] M. Reflojos, «El periodico,» 04 septiembre 2020. [En línea]. Available: <https://www.elperiodico.com/es/activos/innovadores/20200904/pandemia-demandaagilidad-metodologias-agile-empresas-8098246>.
- [3] Tecnologia, «Intra Med,» 31 julio 2020. [En línea]. Available: <https://www.intramed.net/contenidover.asp?contenidoid=96511>.
- [4] J. Hajek, «The Necessity of Interdisciplinary Software Development for Building Viable Research Platforms: Case Study in Automated Drug Delivery in Diabetes,» SIGITE, Chicago, 2020.
- [5] J. Navarro, «Close2U: An App for Monitoring Cancer Patients with Enriched Information from Interaction Patterns,» Hindawi, Zaragoza, 2020.
- [6] H. Tinoco, «Developing a Design Phase for a Mentoring Mobile App,» TEEM, Salamanca, 2020.
- [7] T. A. G. e.-L. P. f. E. Training, «Matevz Pesek,» IEEE Access, Ljubljana, 2020.
- [8] Z. W. Huilong Duan, «Using Goal-Directed Design to Create a Mobile Health App to,» JMIR MHEALT AND UNHEALT, Yinchuan, China, 2020.
- [9] C. C. Soumink Mohian, «Doodle2App: Native App Code by Freehand UI Sketching,» International Conference on Mobile Software Engineering and Systems, Arlington, Texas, 2020.
- [10] C. a. M. R. a. B. J. a. O. M. a. Y. K. a. D. J. O'Grady, «A mobile health approach for improving outcomes in suicide prevention (SafePlan),» HSE Mental Health Services, 2020.
- [11] D.-M. Nguyen, «Automated Test Input Generation via Model Inference Based on User Story and Acceptance Criteria for Mobile Application Development,» WorldScientific, Vietnam, 2020.

- [12] V.-V. Fireteanu, «Agile Methodology Advantages when delivering Internet of Things projects,» Bucharest, 2020.
- [13] D. Martinez, «An Agile-Based Integrated Framework for Mobile Application Development Considering Ilities,» IEEE Access, Quito, 2020.
- [14] A. Olaya, «CardioResyncApp: A mobile phone application for recording research data in Cardiology [CardioResyncApp: Un aplicativo móvil para recolectar datos de investigación en Cardiología],» Revista Colombiana, Bogotá, 2020.
- [15] J. d. Souza, «A Mobile Serious Game About the Pandemic (COVID-19 - Did You Know?): Design and Evaluation Study,» JMIR SERIOUS GAMES, Belo Horizonte, Brazil, 2020.
- [16] M. G. D. P. Evelyn Alvarez, «A software to prevent delirium in hospitalised older adults: development and feasibility assessment,» Universidad de Santiago de Chile, Santiago de Chile, 2020.
- [17] E. L. H. Alexi Delgado, «Design of a Mobile Application for the School Enrollment Process in Order to Prevent Covid-19,» *ASTES*, vol. 5, n° 6, pp. 1042-1046, 2020.
- [18] R. E. R. Viany Utami Tjhin, «Scrum to suport application development project for online learning,» de *In 2020 The 8th International Conference on Information Technology*, Xi'an, China, 2020.
- [19] J. W. Paloma Chausa, «EmERGE mHealth Platform: Implementation and Technical Evaluation of aDigital Supported Pathway of Care for Medically Stable HIV,» *INTERNATIONAL JOURNAL OF ENVIRONMENTAL RESEARCH AND PUBLIC HEALTH*, vol. 18, 2021.
- [20] K. A. S. C. Juniero Neyra, «Multiplatform system to improve tourist services in the city of trujillo [Sistema Multiplataforma para mejorar los Servicios Turísticos en la ciudad de Trujillo],» de *18th LACCEI International Multi-Conference for Engineering, Education, and Technology*, Trujillo, Perú, 2020.

- [21] I. D. S. R F Rahman, «Realtime Notifications on Visitor Tracking Systems Using Android and Arduino,» The Electrochemical Society, Indonesia, 2020.
- [22] K. Sutham, «Thailand medical mobile application for patients triage base on criteria based dispatch protocol,» BMC Medical Informatics and Decision Making, Thailandia, 2020.
- [23] A. N. R Wahdiniwaty, «Application Information System Smart Parking Based on Smartphone,» The electrochemical Society, 2020.
- [24] M. Orellana, «Systems dynamics and serious video games in an appropriation strategy of a decision support system for small livestock farmers,» Unisersity of the Balearic Islands, Palma, España, 2020.
- [25] F. Q. Khan, «A comparative analysis of RAD and agile technique for management of computing graduation projects,» *Computers, Materials and Continua*, vol. 64, n° 2, pp. 777-796, 2020.
- [26] R. V. Diana Isabel Ornoñez, «Mobile application to improve tourist flow in trujillo [Aplicación Móvil para Mejorar el Flujo Turístico en Trujillo],» Universidad César Vallejo, Trujillo, Perú, 2020.
- [27] M. F. Fahad Altuwajri, «Investigating agile adoption in saudi arabian mobile application development,» Qassim Unisersity, Buraydah, Saudi Arabia, 2020.

6.APÉNDICES

APÉNDICE 1 EJECUCIÓN DE SCRIPTS DE BÚSQUEDA

The screenshot shows the Web of Science search interface. At the top, there are search filters for citation indices: Social Sciences Citation Index (SSCI) -2000-presente, Arts & Humanities Citation Index (A&HCI) -2000-presente, and Emerging Sources Citation Index (ESCI) -2015-presente. Below this is a section titled "Historial de búsqueda:" which contains a table with search history. The table has columns for "Conjunto", "Resultados", and "Editar conjuntos". The first row shows a search with 15 results, with a query string: ALL=(university OR companies) AND ALL=("software development process" OR "agile methodology" OR "development methodology") AND ALL=("mobile" OR "mobile application"). The interface also includes buttons for "Guardar historial/Crear alerta" and "Abrir historial guardado".

Conjunto	Resultados	Editar conjuntos	Combinar conjuntos	Eliminar conjuntos
# 1	15 ALL=(university OR companies) AND ALL=("software development process" OR "agile methodology" OR "development methodology") AND ALL=("mobile" OR "mobile application") Índices=SCI-EXPANDED, SSCI, A&HCI, ESCI Período de tiempo=2020-2021	Editar	<input type="radio"/> AND <input type="radio"/> OR Combinar	Seleccionar todo Eliminar

FIGURA 3
BASE DE DATOS WEB OF SCIENCE

The screenshot shows the IEEE Xplore search interface. At the top, there is a search bar with the text "All" and a search button. Below the search bar, there is a section titled "Showing 1-25 of 50 for" followed by a complex query string: (((("All Metadata":university OR companies) AND "All Metadata":mobile OR mobile application) AND "All Metadata":delivery OR "home delivery") AND "All Metadata":"software development process" OR "agile methodology" OR "development methodology") *). Below the query string, there are filters applied: 2020 - 2021. There are also checkboxes for "Conferences (40)", "Journals (6)", "Early Access Articles (3)", and "Magazines (1)". The interface includes a "Show" section with "All Results" selected and "Open Access Only" unselected. The search results are sorted by "Relevance".

FIGURA 4
BASE DE DATOS IEEE Xplore

APÉNDICE 2

IMPORTACION DE LOS ARTICULOS UTILIZANDO PARSIFAL.

Source	Imported Studies
ACM Digital Library	89
IEEE Digital Library	50
ISI Web of Science	15
Scopus	154

FIGURA 5
IMPORTACION DE ARTICULOS EN PARSIFAL

Bibtex Key	Title	Author	Journal	Year	Added by	Added at	Status
ISI:000531558900004	Automated Test Input Generation via Model Inference Based on User Story and Acceptance Criteria for Mobile Application Development	Duc-Man Nguyen and Quyet-Thang Huynh and Nhu-Hang Ha and Thanh-Hung Nguyen	INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING	2020	AronCamacho	06 Jun 2021 17:42:12	Accepted
ISI:000530827300017	An Agile-Based Integrated Framework for Mobile	Martinez, Danilo and Ferre, Xavier and Guerrero, Graciela and Juristo, Natalia	IEEE ACCESS	2020	AronCamacho	06 Jun 2021 17:42:12	Accepted

FIGURA 6
IMPORTACION DE ARITUCLOS EN PARSIFAL

APÉNDICE 3

REALIZACIÓN DE ANÁLISIS DE LOS ARTÍCULOS RELACIONADOS.

Action: 0 of 308 selected

Show: All Accepted Rejected Unclassified Duplicated

<input type="checkbox"/>	Bibtex Key	Title	Author	Journal	Year	Added by	Added at	Status
<input type="checkbox"/>	ISI:000531558900004	Automated Test Input Generation via Model Inference Based on User Story and Acceptance Criteria for Mobile Application Development	Duc-Man Nguyen and Quyet-Thang Huynh and Nhu-Hang Ha and Thanh-Hung Nguyen	INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING	2020	AronCamacho	06 Jun 2021 17:42:12	Accepted
<input type="checkbox"/>	ISI:000530827300017	An Agile-Based Integrated Framework for Mobile Application Development Considering Iltities	Martinez, Danilo and Ferre, Xavier and Guerrero, Graciela and Juristo, Natalia	IEEE ACCESS	2020	AronCamacho	06 Jun 2021 17:42:12	Accepted
<input type="checkbox"/>	Ribaut2020	Theory-driven development of a medication adherence intervention delivered by eHealth and transplant team in allogeneic stem cell transplantation: The SMILe implementation science project	Ribaut, J. and Leppla, L. and Teynor, A. and Valenta, S. and Dobbels, F. and Zullig, L.L. and De Geest, S. and Beckmann, S. and Mielke, J. and Schmid, A. and Duerinckx, N. and Heidegger, P. and Fürmann, M. and Neupert, D. and Rockstein, D. and Werlitz, V. and Fürmann, M. and Schulz, T. and Lemcke, M. and Schumacher, V. and Zeiser, R. and Engelhardt, M. and Hasemann, M. and Kaier, K. and Gerull, S. and Passweg, J. and Ulrich, A. and Grossmann, F. and Bolliger, D. and Reitwiessner, S. and Degen, S. and Schonfeld, S. and Senft, Y. and Maier, B. and Shultis, C.	BMC Health Services Research	2020	AronCamacho	06 Jun 2021 17:43:40	Rejected

FIGURA 7
ANÁLISIS DE TÍTULO Y RESUMEN

<input type="checkbox"/>	ISI:000624817600001	Agile Beeswax: Mobile App Development Process and Empirical Study in Real Environment	Alrabaiah, Hazem Abdelkarim and Medina-Medina, Nuria	SUSTAINABILITY	2021	AronCamacho	06 Jun 2021 17:42:12	Duplicated
<input type="checkbox"/>	Beschastnikh2020	Visualizing Distributed System Executions	Beschastnikh, I. and Liu, P. and Xing, A. and Wang, P. and Brun, Y. and Ernst, M.D.	ACM Transactions on Software Engineering and Methodology	2020	AronCamacho	06 Jun 2021 17:43:40	Rejected
<input type="checkbox"/>	Borg2020246	Video Game Development in a Rush: A Survey of the Global Game Jam Participants	Borg, M. and Garousi, V. and Mahmoud, A. and Olsson, T. and Stalberg, O.	IEEE Transactions on Games	2020	AronCamacho	06 Jun 2021 17:43:40	Rejected
<input type="checkbox"/>	Duan2020	Using goal-directed design to create a mobile health app to improve patient compliance with hypertension self-management: Development and deployment	Duan, H. and Wang, Z. and Ji, Y. and Ma, L. and Liu, F. and Chi, M. and Deng, N. and An, J.	JMIR mHealth and uHealth	2020	AronCamacho	06 Jun 2021 17:43:40	Accepted
<input type="checkbox"/>	Raharjana202153811	User Stories and Natural Language Processing: A Systematic Literature Review	Raharjana, I.K. and Siahaan, D. and Fatichah, C.	IEEE Access	2021	AronCamacho	06 Jun 2021 17:43:40	Rejected
<input type="checkbox"/>	Weichbroth202055563	Usability of mobile applications: A systematic literature study	Weichbroth, P.	IEEE Access	2020	AronCamacho	06 Jun 2021 17:43:40	Rejected
<input type="checkbox"/>	Johnson2020	Usability methods and attributes reported in usability studies of mobile apps for health care education: Protocol for a scoping review	Johnson, S.G. and Potrebny, T. and Larun, L. and Ciliska, D. and Olsen, N.R.	JMIR Research Protocols	2020	AronCamacho	06 Jun 2021 17:43:40	Rejected

FIGURA 8
ANÁLISIS de TITULO Y RESUEMEN

APÉNDICE 4

LECTURA CRÍTICA Y CONTROL DE CALIDAD

ANÁLISIS DE CALIDAD

Protocol
Quality Assessment Checklist
Data Extraction Form

Quality Assessment Checklist

Questions

Answers

Score

Questions

^	¿El estudio presenta al menos una metodología de desarrollo de software para desarrollo de aplicaciones móviles?	edit remove
v		
^	¿El estudio indica los criterios de selección de la metodología usada?	edit remove
v		
^	¿El estudio muestra las fases de desarrollo de la aplicación?	edit remove
v		
^	¿El estudio indica el Sistema Operativo para el cual va a ser desarrollado la aplicación?	edit remove
v		

+ Add Question

Answers

Description	Weight	
Yes	1.0	edit remove
Partially	0.5	edit remove
No	0.0	edit remove

+ Add Answer

FIGURA 9
CUESTIONARIO DE CALIDAD

APÉNDICE 5: CUESTIONARIO DE CALIDAD

1. Search 2. Import Studies 3. Study Selection 4. Quality Assessment 5. Data Extraction 6. Data Analysis

Quality Assessment

Detailed Summary

Show: All Done Pending Score higher than 2.5 Score lower or equal to 2.5 Order by: Title (a - z) ▾

To answer the form you may click on the desired answer on the following tables.

A comparative analysis of RAD and agile technique for management of computing graduation projects (2020) 3.5			
¿El estudio presenta al menos una metodología de desarrollo de software para desarrollo de aplicaciones móviles?	Yes	Partially	No
¿El estudio indica los criterios de selección de la metodología usada?	Yes	Partially	No
¿El estudio muestra las fases de desarrollo de la aplicación?	Yes	Partially	No
¿El estudio indica el Sistema Operativo para el cual va a ser desarrollado la aplicación?	Yes	Partially	No

Agile Accelerator Program: From Industry-Academia Collaboration to Effective Agile Training (2020) 0.5			
¿El estudio presenta al menos una metodología de desarrollo de software para desarrollo de aplicaciones móviles?	Yes	Partially	No
¿El estudio indica los criterios de selección de la metodología usada?	Yes	Partially	No
¿El estudio muestra las fases de desarrollo de la aplicación?	Yes	Partially	No
¿El estudio indica el Sistema Operativo para el cual va a ser desarrollado la aplicación?	Yes	Partially	No

FIGURA 10
CUESTIONARIO DE CALIDAD

Quality Assessment

Detailed **Summary**

Green means higher than cutoff score. Red means lower or equal to the cutoff score.

Title	Quality Score
Automated Test Input Generation via Model Inference Based on User Story and Acceptance Criteria for Mobile Application Development	4.0
An Agile-Based Integrated Framework for Mobile Application Development Considering Illities	4.0
EmERGE mHealth Platform: Implementation and Technical Evaluation of a Digital Supported Pathway of Care for Medically Stable HIV	3.0
Realtime Notifications on Visitor Tracking Systems Using Android and Arduino	3.5
Thailand medical mobile application for patients triage base on criteria based dispatch protocol	3.5
CardioResyncApp: A mobile phone application for recording research data in Cardiology [CardioResyncApp: Un aplicativo móvil para recolectar datos de investigación en Cardiología]	3.5
The Necessity of Interdisciplinary Software Development for Building Viable Research Platforms: Case Study in Automated Drug Delivery in Diabetes	4.0
Developing a Design Phase for a Mentoring Mobile App	3.0
Close2U: An App for Monitoring Cancer Patients with Enriched Information from Interaction Patterns	3.5
Using goal-directed design to create a mobile health app to improve patient compliance with hypertension self-management: Development and deployment	4.0
The Development of Web-Based Correspondence Information Systems in University Using Scrum	1.5
Systems dynamics and serious video games in an appropriation strategy of a decision support system for small livestock farmers	3.0
Ramping up customer-centric modular design projects: Mobile app development for pandemic relief	2.5

FIGURA 11
Estudios evaluados por el Cuestionario de Calidad

Anexo 2: Pruebas de Escenarios

Matriz de escenarios I

Escenario	tipo	Modulo	Requerimiento	Detalle del proceso
Registro de un usuario en el sistema	post	RegistrarUsuario	nombre, apellidos, dirección, teléfono, correo electrónico, contraseña.	Permitir que los usuarios clientes se registren para que puedan tener su cuenta.
Ingreso de un usuario en el sistema	post	IdentificarUsuario	Correo, password,tipo	Permite que los usuarios clientes y usuarios restaurantes puedan iniciar sesión en la aplicación.
Realizar pedido por un cliente	post	RealizarPedido	producto,precio,cantidad,direccion,tipoEntrega	Permitir que los clientes puedan realizar pedidos a los restaurantes disponibles en la aplicación.
Mostrar los restaurantes disponibles	get	VisualizarRestaurantes	nombreRestaurante,valoracion,direccion	Permitir que los usuarios clientes puedan ver el listado de todos los restaurantes y menús disponibles.
Modificar datos de una cuenta cliente	post	ModificarCuenta	nombre,correo,telefono,direccion,fotoPerfil	El cliente ingresa a su perfil y puede modificar su cuenta dando clic en el botón "Modificar cuenta".
Mostrar historial de pedidos	get	HistorialPedidos	nombreRestaurante,totalPedido,direccion,cantidadProductos	El usuario cliente ingresa a la pestaña historial y puede observar el historial de sus pedidos.
Recibir notificación sobre un pedido	get	RecibirNotificacion	detalles pedido, menú, precio, nombres cliente, dirección, teléfono.	Al registrarse algún pedido a determinado restaurante, este recibirá una notificación en la aplicación.
Gestión de los productos de los restaurantes	post	GestionarCatalogo		El usuario restaurante ingresa a la pestaña de agregar productos y puede agregar, modificar o eliminar su menú.
Gestión de las ofertas de los restaurantes	post	GestionarCatalogo		El usuario restaurante ingresa a la pestaña de agregar ofertas y puede agregar, modificar o eliminar sus ofertas.

Agregar un menú al catálogo del restaurante	post	AgregarProducto	nombreProducto,descripcionProducto,precioProducto,imagenProducto	El usuario restaurante ingresa a la pestaña de agregar producto y puede agregar productos dando clic en el icono comida.
Modificar menú del catálogo del restaurante	post	ModificarProducto	nombreProducto,descripcionProducto,precioProducto,imagenProducto (no es necesario modificar todos los campos)	El usuario restaurante selecciona un producto puede modificar el producto dando clic en el botón modificar.
Agregar oferta de restaurante	post	AgregarOferta	nombreOferta,descripcionOferta,precioOferta,imagenOferta,tiempoOferta	El usuario restaurante ingresa a la pestaña de agregar oferta y puede agregar ofertas dando clic en el icono oferta.
Modificar oferta del restaurante	post	ModificarOferta	nombreOferta,descripcionOferta,precioOferta,imagenOferta,tiempoOferta	El usuario restaurante selecciona un producto puede modificar el producto dando clic en el botón modificar.
Gestión de los restaurantes por parte del administrador	post	GestionarRestaurante	nombreRestaurante,valoracion,direccion,telefono,correo,imagenRestaurante	El administrador agrega los restaurantes desde la base de datos directamente

Matriz de escenarios II

ID	MODULO	REQUERIMIENTO	DETALLE DEL PROCESO	RESULTADO ESPERADO	Fecha		Resultado	
					Fecha	Resultado	Fecha	Resultado
1	RegistrarUsuario	nombre, apellidos, dirección, teléfono, correo electrónico, contraseña.	Permitir que los usuarios clientes se registren para que puedan tener su cuenta.	El usuario ingresado se registrará correctamente	2021-08-02	usuario registrado correctamente		
2	RegistrarUsuario	nombre, apellidos, dirección, teléfono, correo electrónico, contraseña.	Se tratará de registrarse con el mismo correo nuevamente	el usuario ingresado ya está registrado	2021-08-02	mensaje de usuario ya registrado		
3	RegistrarUsuario	nombre, apellidos, dirección, teléfono, correo electrónico, contraseña.	No se llenará todos los campos del formulario	No se puede crear la cuenta debido a los campos vacíos	2021-08-02	mensaje de usuario no registrado		
4	IdentificarUsuario	Correo, password	Permite que los usuarios clientes puedan iniciar sesión en la aplicación.	Ingreso a la aplicación móvil exitosa	2021-08-02	ingreso exitoso		
5	IdentificarUsuario	Correo, password,	Permite que los usuarios restaurante puedan iniciar sesión en la aplicación.	Ingreso a la aplicación móvil exitosa	2021-08-02	ingreso exitoso		
6	RealizarPedido	producto,precio,cantidad,direccion,tipoEntrega	Permitir que los clientes puedan realizar pedidos a los restaurantes disponibles en la aplicación.	Pedido realizado con éxito	2021-08-02	Pedido registrado exitosamente		
7	VisualizarRestaur antes	nombreRestaurante,valoracion,direccion	Permitir que los usuarios clientes puedan ver el listado de todos los restaurantes y menús disponibles.	Se debe visualizar todos los restaurantes registrados en la pantalla de inicio	2021-08-02	Los restaurantes se pueden visualizar exitosamente		
8	ModificarCuenta	nombre,correo,telefono,direccion,fotoPerfil	El cliente puede modificar su cuenta dando clic en el botón "Modificar cuenta".	Los datos modificados deben reflejarse luego de guardar los cambios realizados	2021-08-02	Los resultados tardan en reflejarse, pero se pueden evidenciar luego de un tiempo		
9	HistorialPedidos	nombreRestaurante,totalPedido,direccion,cantidadProductos	El usuario cliente ingresa a la pestaña historial y puede	Los pedidos previamente	2021-08-02	Los pedidos realizados		

			observar el historial de sus pedidos	realizados deben poder visualizarse		previamente se pueden visualizar exitosamente
10	RecibirNotificación	detalles pedido, menú, precio, nombres cliente, dirección, teléfono.	Al registrarse algún pedido a determinado restaurante, este recibirá una notificación en la aplicación.	El restaurante recibe una notificación acerca del pedido solicitado	2021-08-02	El restaurante recibe las notificaciones de sus pedidos con éxito
11	AgregarProducto	nombreProducto,descripcionProducto,precioProducto,imagenProducto	El usuario restaurante puede agregar productos dando clic en el icono comida.	Producto agregado al menú del restaurante	2021-08-02	El producto puede ser agregado sin inconvenientes
12	ModificarProducto	nombreProducto,descripcionProducto,precioProducto,imagenProducto	El usuario restaurante puede modificar el producto dando clic en el botón modificar.	Información del producto modificada	2021-08-02	La información del producto es modificada con éxito
13	AgregarOferta	nombreOferta,descripcionOferta,precioOferta,imagenOferta,tiempoOferta	El usuario restaurante puede agregar ofertas dando clic en el icono oferta.	Oferta agregada al menú del restaurante	2021-08-02	La oferta fue agregada sin inconvenientes
14	ModificarOferta	nombreOferta,descripcionOferta,precioOferta,imagenOferta,tiempoOferta	El usuario restaurante puede modificar el producto dando clic en el botón modificar.	Información de la oferta modificada	2021-08-02	La información de la oferta fue modificada con éxito

Anexo 3: Pruebas de Aceptación

Organismo	Universidad Nacional de Loja 		
Proyecto	APLICACIÓN MÓVIL PARA SERVICIOS DE COMIDA		
Entregable	Pruebas de aceptación		
Producto a Evaluar	Aplicación Móvil		
Aprobado por	Andy Aron Camacho Herrera	Fecha de Aprobación	2021/08/4

Nro	Actividad	Cumple	no Cumple	Observación
1	La aplicación móvil es intuitiva para el cliente	✓		
2	La aplicación móvil es amigable con el cliente	✓		
3	La aplicación móvil es eficiente al momento de realizar los pedidos	✓		
4	La aplicación móvil contiene tiempos de respuesta aceptables	✓		
5	La aplicación móvil cumple con los cambios sugeridos por el cliente	✓		
6	La aplicación móvil cumple con los requisitos y módulos previamente definidos	✓		



.....
Andy Aron Camacho Herrera
 (Cliente)

.....
Ángel Ricardo Capa
 (Tester)

Anexo 4: Prototipos de la aplicación

Los prototipos de la aplicación de prueba se pueden obtener en los siguientes enlaces:

- **PrototipoV.1.0:**

<https://www.figma.com/file/b1N2wZlvsYgijiO2l4kAt/Proto1?node-id=0%3A1>

- **PrototipoV.2.0:**

<https://www.figma.com/file/CRarh8n2yVdplezucqXQ2D/Prototipo-Version2?node-id=0%3A1>

- **Código de la aplicación:**

<https://drive.google.com/drive/folders/1mhyDRt-YWVMJFjU-eOglaDMHzYdfmqFj>

Anexo 5: Certificado de traducción del resumen

CERTIFICADO DE TRADUCCIÓN

La **Sra. María Belén Castillo Granda**, identificada con número de cédula **1105673204**, graduada: **“Suficiencia en inglés nivel B2”**.

CERTIFICA:

Que el texto traducido al idioma inglés que compone el **Resumen** del Trabajo de Titulación denominado: **“Estudio comparativo de metodologías de desarrollo de software para aplicaciones móviles / Comparative study of software development methodologies for mobile applications”** correspondiente al **Sr. Andy Aron Camacho Herrera** con numero de cédula **1150061032**, fue realizado y verificado bajo mi supervisión.

Eso es todo en cuanto puedo indicar en honor a la verdad, facultando al interesado hacer uso del presente documento para los fines que crea pertinentes.

Loja, 4 de mayo de 2022



.....
María Belén Castillo Granda

C.I: 1105673204

Celular: 0967656060