



UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

Metodología para la evaluación del rendimiento de aplicaciones
móviles

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERO DE SISTEMAS**

AUTORES:

Baldarrago Pilco, Romario Alberto (0000-0002-4326-0799)

Mendoza Huaman, Anderson Christofer (0000-0001-8777-0427)

ASESOR:

Dr. Hilario Falcón, Francisco Manuel (0000-0003-3153-9343)

LÍNEA DE INVESTIGACIÓN:

Sistema de Información y Comunicaciones

LIMA – PERÚ

2021

Dedicatoria

A nuestros padres por su ayuda incondicional y por ser el principal apoyo a lo largo de nuestra formación profesional.

Agradecimiento

Al Dr. Francisco Manuel Hilario Falcón, docente del curso, por la orientación, el seguimiento y la supervisión continua durante el desarrollo de nuestra investigación y al Dr. Emigdio Antonio Alfaro Paredes, por sus recomendaciones, sugerencias y enseñanzas.

Índice de contenidos

I. INTRODUCCIÓN.....	1
II. MARCO TEÓRICO	8
III. METODOLOGÍA.....	17
3.1 Tipo y diseño de investigación.....	18
3.2 Variables y operacionalización	18
3.3 Población, muestra y muestreo	19
3.4 Técnicas e instrumentos de recolección de datos.....	20
3.5 Procedimientos.....	20
3.6 Método de análisis de datos	21
3.7 Aspectos éticos	22
IV. RESULTADOS	23
V. DISCUSIÓN	39
VI. CONCLUSIONES.....	42
VII. RECOMENDACIONES.....	44
REFERENCIAS	46
ANEXOS	

Índice de tablas

Tabla 1: Estadísticos descriptivos indicador uso del CPU	24
Tabla 2: Prueba de normalidad indicador uso del CPU	25
Tabla 3: Estadísticos descriptivos indicador uso de la memoria RAM	25
Tabla 4: Prueba de normalidad indicador uso de la memoria RAM	26
Tabla 5: Estadísticos descriptivos indicador uso de la batería	27
Tabla 6: Prueba de normalidad indicador uso de la batería	28
Tabla 7: Estadísticos descriptivos indicador tiempo de respuesta.....	28
Tabla 8: Prueba de normalidad indicador tiempo de respuesta.....	29
Tabla 9: Estadísticos descriptivos indicador conexión a la red.....	30
Tabla 10: Prueba de normalidad indicador conexión a la red.....	31
Tabla 11: Estadísticos descriptivos indicador carencia de seguridad	31
Tabla 12: Prueba de normalidad indicador carencia de seguridad	32
Tabla 13: Prueba de Wilcoxon indicador uso del CPU	33
Tabla 14: Prueba de Wilcoxon indicador uso de la memoria RAM	33
Tabla 15: Prueba de Wilcoxon indicador uso de la batería.....	33
Tabla 16: Prueba de hipótesis – Uso de recursos (Uso del CPU)	33
Tabla 17: Prueba de hipótesis – Uso de recursos (Uso de la memoria RAM)	33
Tabla 18: Prueba de hipótesis – Uso de recursos (Uso de la batería).....	34
Tabla 19: Prueba de Wilcoxon indicador tiempo de respuesta	34
Tabla 20: Prueba de hipótesis – Tiempo de respuesta	35
Tabla 21: Prueba de Wilcoxon indicador conexión a la red	35
Tabla 22: Prueba de hipótesis – Conexión a la red.....	35
Tabla 23: Prueba de Wilcoxon indicador carencia de seguridad	36
Tabla 24: Prueba de hipótesis – Carencia de seguridad	36
Tabla 25: Resumen de las pruebas de hipótesis.....	37
Tabla 26: Matriz de operacionalización de variables	54
Tabla 27: Matriz de consistencia.....	55
Tabla 28: Descripción de los parámetros de rendimiento por indicadores.....	70
Tabla 29: Ficha de recopilación de datos – FORM-01 del indicador uso del CPU.....	72
Tabla 30: Ficha de recopilación de datos – FORM-02 del indicador uso de memoria RAM	75
Tabla 31: Ficha de recopilación de datos – FORM-04 del indicador uso de la batería	77
Tabla 32: Ficha de recopilación de datos – FORM-04 del indicador tiempo de respuesta.....	78
Tabla 33: Ficha de recopilación de datos – FORM-05 del indicador conexión a la red.....	80
Tabla 34: Ficha de recopilación de datos – FORM-06 del indicador carencia de seguridad.....	86

Índice de figuras

Figura 1. Arquitectura utilizada en la investigación.....	57
Figura 2. Fases de la metodología para la evaluación del rendimiento de aplicaciones móviles.	58
Figura 3. Estructura de las fases de la metodología para la evaluación del rendimiento de aplicaciones móviles.	59
Figura 4. Arquitectura de la metodología.	61
Figura 5. Estructura de los procesos de la metodología para la evaluación del rendimiento de aplicaciones móviles.....	63
Figura 6. Características y funcionalidades necesarias de las aplicaciones que fueron sometidas a pruebas.....	87
Figura 7. Herramientas utilizadas para las pruebas de uso de recursos, tiempo de respuesta, conexión a la red y carencia de seguridad (Device Info HW, Simple System Monitor y CPU/GPU Meter & Notification).	88
Figura 8. Identificación de las características del dispositivo utilizando la herramienta Device Info HW).....	89
Figura 9. Instalación de las aplicaciones móviles.	90
Figura 10. Ejecución de pruebas de rendimiento uso de CPU.	91
Figura 11. Ejecución de pruebas de rendimiento uso de memoria RAM.	92
Figura 12. Ejecución de pruebas de rendimiento uso de la batería.	92
Figura 13. Ejecución de pruebas de tiempo de respuesta.....	93
Figura 14. Archivos preparados para la ejecución de la prueba de conexión a la red.	94
Figura 15. Ejecución de prueba de rendimiento de conexión a la red.	94
Figura 16. Ejecución de prueba de carencia de seguridad.....	95

Índice de abreviaturas

ANP: Analytic Network Process

CIEI: Centro de Investigación y Evaluación Institucional

CPU: Central Processing Unit

DCU: Diseño Centrado en el Usuario

DEMATEL: Decision Making Trial and Evaluation Laboratory

GPS: Global Positioning System

GPU: Graphics Processing Unit

HTML: HyperText Markup Language

MCDA: Multi- Criteria Decision-Analysis

MERAM: Metodología para evaluar el rendimiento de aplicaciones móviles

OWASP: Open Web Application Security Project. Es Organización sin fines de lucro dedicada a la seguridad de aplicaciones e impulsada por una comunidad abierta de profesionales de la seguridad de casi todos los países del mundo.

Q2M: Librería para evaluar la calidad de un software móvil

RAM: Random Access Memory

WWWC: World Wide Web Consortium

Resumen

Reflexionando lo primordial que es la calidad de un software en el desarrollo y el uso que emplean los usuarios con las aplicaciones móviles, en este proyecto de investigación se muestra un análisis persuasivo comparativo de dos tipos de aplicaciones móviles, aplicando una metodología con la finalidad de evaluar su rendimiento.

El problema de la investigación fue que no se ha encontrado una metodología que permita evaluar el rendimiento de las aplicaciones móviles con respecto al uso de recursos, tiempo de respuesta, conexión a la red y carencia de seguridad lo que genera un problema al elegir el tipo de aplicación móvil más adecuado. La presente investigación, tuvo como objetivo general: elaborar una metodología para evaluar el rendimiento de las aplicaciones móviles MERAM y aplicarla con el fin de determinar el rendimiento de una aplicación móvil nativa versus la misma aplicación desarrollada con tecnología híbrida teniendo en cuenta el consumo de recursos, tiempo de respuesta, conexión a la red y carencia de seguridad.

Se recolectó y analizó información de artículos científicos y libros para seleccionar los métodos, técnicas y herramientas más adecuados para la evaluación de software móvil. Con ello se elaboró la metodología MERAM que está conformada por 7 procesos: (a) seleccionar las aplicaciones móviles, (b) preparar el ambiente de pruebas, (c) realizar pruebas de uso de recursos, (d) realizar pruebas de tiempo de respuesta, (e) realizar pruebas de conexión a la red, (f) realizar pruebas de carencia de seguridad y (g) analizar los datos. Además, la metodología MERAM para la ejecución de las pruebas utiliza software libre, tales como: Simple System Monitor, Device info HW, CPU/GPU Meter & Notification e Immuniweb

Palabras clave: Metodología, rendimiento, aplicaciones móviles, aplicaciones híbridas, aplicaciones nativas

Abstract

Reflecting the importance of software quality in the development and use of mobile applications by users, this research project shows a comparative persuasive analysis of two types of mobile applications, applying a methodology in order to evaluate their performance.

The problem of the research was that no methodology has been found to evaluate the performance of mobile applications with respect to the use of resources, response time, network connection and lack of security, which generates a problem when choosing the most appropriate type of mobile application. The general objective of this research was to develop a methodology to evaluate the performance of MERAM mobile applications and apply it to determine the performance of a native mobile application versus the same application developed with hybrid technology, taking into account resource consumption, response time, network connection and lack of security.

Information from scientific articles and books was collected and analyzed to select the most appropriate methods, techniques and tools for mobile software evaluation. This led to the development of the MERAM methodology, which is made up of 7 processes: (a) selecting the mobile applications, (b) preparing the test environment, (c) performing resource usage tests, (d) performing response time tests, (e) performing network connection tests, (f) performing security lack tests and (g) analyzing the data. In addition, the MERAM methodology for the execution of the tests uses free software, such as: Simple System Monitor, Device info HW, CPU/GPU Meter & Notification and Immuniweb.

Keywords: Methodology, performance, mobile applications, hybrid applications, native applications.

I. INTRODUCCIÓN

En el presente capítulo se presenta la realidad problemática, la formulación del problema, la justificación, objetivos e hipótesis.

En todo el mundo existe millones de aplicaciones que son usadas diariamente con fines de entretenimiento, comunicación, trabajo, salud, compras, juegos, etc. Estas aplicaciones son usadas por usuarios que esperan sean rápidas y que no usen demasiados recursos de sus dispositivos. Sahu (2018) indicó que las aplicaciones móviles son muy importantes en la vida diaria de las personas ya que las mantienen conectadas con amigos, familiares y compañeros de trabajo además resalta a las aplicaciones con acceso a la red (p. 44).

Las aplicaciones móviles son cada vez más usadas por lo tanto todos los días se suman nuevas aplicaciones a las tiendas Google Play y App Store que son las más populares, pero no todas las aplicaciones son exitosas. Angelova (2019) señaló que es importante que el producto tenga un alto rendimiento, además mencionó que una de las principales razones por las que un usuario elimina una aplicación de su dispositivo móvil es por el gran tamaño y el alto consumo de almacenamiento en su dispositivo (p. 856).

Al respecto Thomas et al. (2018) mencionaron distintos tipos de aplicaciones, sin embargo se tomará en cuenta dos de ellas, nativas e híbridas. Las aplicaciones móviles nativas así empleadas adquieren un grupo de características de gran ventaja, lo que sobresale es su acceso a las distintas capacidades del dispositivo móvil (GPS, agenda, cámara, micrófono, entre otras), su alto rendimiento, la oportunidad de trabajar sin internet y ejecutar en segundo plano avisando al usuario solo si se requiere de su atención. Estas aplicaciones tienen la posibilidad de ser distribuidas en cualquier tipo de tienda online como Google Play, App Store, Huawei Store, entre otros. Estos son muy buenos en rendimiento y apariencia, pero costosos en desarrollo (p. 589).

Por otro lado, Díaz, Herrera y Rosenzvaig (2019) mencionaron que las aplicaciones móviles de forma híbrida emplean la metodología web, pero estas se ejecutan dentro de una caja web (webview), forma parte de la tecnología nativa, en donde se da la oportunidad de acceder a distintos tipos de recursos del dispositivo móvil, En pocas palabras lo híbrido adquiere cada una de las

ventajas de las otras formas móviles como lo son nativa, web responsivas, interpretadas, generadas por compilación cruzada y entre otras. Si se habla del costo y tiempo del desarrollo, este tipo de aplicación vendría siendo la mejor opción, sin embargo, si se toma en cuenta el rendimiento e interfaz, esta no podría ser una opción viable para que las empresas puedan adquirir y ser entregadas a sus usuarios a través de las tiendas virtuales, puesto que carecen de ello. (p. 664).

Cárdenas et al. (2021) explicó que la tecnología artificial en las aplicaciones móviles está simbolizada actualmente un poco factible para ser empleada en el mercado, esto debido a su poca aplicación con las especificaciones de las aplicaciones móviles, el elevado nivel de consumo de recursos. Refiriéndose a los intereses de los usuarios en cuanto a la experiencia de uso, no se podría afirmar que una tendencia es mayor o superior que otra, puesto que todas ellas se centran en satisfacer una o distintas áreas determinadas o buscar la satisfacción del usuario (p. 142-143).

Para Chaverri (2017) la justificación es el entorno pertinente para dar argumentación persuasiva a tal elección, lo cual se puede llegar a cabo de distintas formas. Trata de fundamentar la intranquilidad por comprender un aspecto exclusivo de la materialidad social más allá de un gusto intelectual, entregando objetividad a un proceso ciertamente subjetivo (p. 186). Según lo citado por Chaverri, la justificación tendrá un papel importante puesto que este dará una argumentación persuasiva del por qué se realiza esta investigación, más allá de que sea un capricho del investigador.

Para la justificación práctica, en el presente proyecto se esquematiza una metodología de evaluación del rendimiento de aplicaciones móviles, lo cual resolverá el principal problema que vendría a ser el no disponer de secuencias de una metodología que evalúe el mejor rendimiento de aplicaciones móviles y tomar una decisión al elegir un tipo de aplicación en base a resultados comprobados. Como bien lo señaló Delía (2017) se debe ampliar el análisis comparativo del rendimiento de las aplicaciones móviles, teniendo en cuenta el consumo de la batería, el acceso al disco y el tiempo que una aplicación tarda

en renderizar imágenes además recomienda estudiar la experiencia de usuario en las aplicaciones generados por los diferentes enfoques de desarrollo (p. 78).

Para la justificación social se da como aportes conocimiento que facilitará la evaluación del rendimiento de las aplicaciones móviles y servirá como base para futuras investigaciones relacionadas con este tema. Paniagua, Bedoya y Mera (2020) recomendaron que es importante explicar la naturaleza de las pruebas de software para a futuro evitar deserciones y frustraciones de los usuarios al momento de usar las aplicaciones (p. 114).

De manera metodológica se aporta una metodología para evaluar el rendimiento de las aplicaciones móviles para elegir de acuerdo con los requerimientos entre una aplicación nativa o una híbrida ya que en la búsqueda de los antecedentes no se ha encontrado una metodología que sirva como guía para analizar el rendimiento que tienen las aplicaciones móviles. Velásquez et al. (2019) resaltaron que existe la necesidad de hallar estrategias para facilitar la evaluación de las aplicaciones móviles en diferentes tipos de dispositivos (p. 48).

De manera teórica ya que esta investigación busca aportar conocimiento a través de una metodología debido a que en la actualidad no existe una secuencia de pasos para hacer una evaluación del rendimiento de las aplicaciones móviles. Thomas et al. (2017) señalaron que debido a las variadas formas de crear aplicaciones móviles es necesario nuevas metodologías y prácticas que beneficien el desarrollo de la ingeniería de software (p. 429).

Teniendo en cuenta la situación problemática, el problema general de la investigación propuesta fue que no se ha encontrado una metodología que permita evaluar el rendimiento de las aplicaciones móviles con respecto al uso de recursos, tiempo de respuesta, conexión a la red y carencia de seguridad lo que genera un problema al elegir el tipo de aplicación móvil más adecuado. Teniendo como problemas específicos de la investigación los siguientes:

- **PE1:** No se ha encontrado una metodología que permita evaluar el rendimiento de las aplicaciones móviles con respecto al uso de recursos,

lo que genera un problema al elegir el tipo de aplicación móvil más adecuado.

- **PE2:** No se ha encontrado una metodología que permita evaluar el rendimiento de las aplicaciones móviles con respecto al tiempo de respuesta lo que genera un problema al elegir el tipo de aplicación móvil más adecuado.
- **PE3:** No se ha encontrado una metodología que permita evaluar el rendimiento de las aplicaciones móviles con respecto a la conexión a la red, lo que genera un problema al elegir el tipo de aplicación móvil más adecuado.
- **PE4:** No se ha encontrado una metodología que permita evaluar el rendimiento de las aplicaciones móviles con respecto a la carencia de seguridad, lo que genera un problema al elegir el tipo de aplicación móvil más adecuado.

El objetivo general fue elaborar una metodología para evaluar el rendimiento de las aplicaciones móviles y aplicarla con el fin de determinar el rendimiento de una aplicación móvil nativa versus la misma aplicación desarrollada con tecnología híbrida teniendo en cuenta el uso de recursos, el tiempo de respuesta, la conexión a la red y la carencia de seguridad. Los objetivos específicos fueron los siguientes:

- **OE1:** Elaborar una metodología para evaluar el rendimiento de las aplicaciones móviles y aplicarla con el fin de determinar el rendimiento de una aplicación móvil nativa versus la misma aplicación desarrollada con tecnología híbrida teniendo en cuenta el uso de recursos.
- **OE2:** Elaborar una metodología para evaluar el rendimiento de las aplicaciones móviles y aplicarla con el fin de determinar el rendimiento de una aplicación móvil nativa versus la misma aplicación desarrollada con tecnología híbrida teniendo en cuenta el tiempo de respuesta.
- **OE3:** Elaborar una metodología para evaluar el rendimiento de las aplicaciones móviles y aplicarla con el fin de determinar el rendimiento

de una aplicación móvil nativa versus la misma aplicación desarrollada con tecnología híbrida teniendo en cuenta la conexión a la red.

- **OE4:** Elaborar una metodología para evaluar el rendimiento de las aplicaciones móviles y aplicarla con el fin de determinar el rendimiento de una aplicación móvil nativa versus la misma aplicación desarrollada con tecnología híbrida teniendo en cuenta la carencia de seguridad.

Continuando con la investigación se realizó el planteamiento de la hipótesis; cabe resaltar que el modelo de investigación empleado fue con enfoque cuantitativo y diseño no experimental, descriptivo, lo cual muestra que solo se enuncia hipótesis cuando se predice alguna acción como es la investigación.

Conforme las teorías, se encuentran investigaciones que para lograr el objetivo general no acapara en cuenta cierta metodología propuesta lo que impulsó a delimitar la siguiente hipótesis general la cual fue: Haciendo uso de una metodología se determinará que una aplicación nativa tiene mejor rendimiento que una aplicación híbrida con respecto al uso de recursos, tiempo de respuesta, conexión a la red y carencia de seguridad. De igual modo las hipótesis específicas fueron las siguientes:

- **HE1:** Las aplicaciones nativas son más eficientes en cuanto al uso de recursos. Salgado (2018) indicó que el consumo en cuanto a la batería es muy similar en las dos aplicaciones, tanto híbrido como nativo. Sin embargo, concluyó en base a una estadística que realizó de pruebas que el híbrido es el que absorbe mucha más energía que el nativo (p. 99). Además Durán, Peinado y Rosado (2015) indicaron que en cuanto a la utilización del CPU la aplicación híbrida está por arriba del 50%. Este exagerado uso del CPU podría generar problemas en cuanto al rendimiento del sistema e insatisfacción del usuario. Al iniciar la aplicación nativa se logró emplear únicamente 2,5% de CPU, una diferencia abismal de parte de esta (p. 86). Por otro lado Bosnic, Papp y Novak (2016) firmaron un mayor uso de la memoria RAM por parte de aplicaciones híbridas. Estos resultados ya eran esperados, puesto que las aplicaciones híbridas utilizan WebView, que a su vez utilizan una gran cantidad significativa de memoria RAM (p. 3).

- **HE2:** Las aplicaciones nativas son más eficientes en cuanto al tiempo de respuesta. Durán, Peinado y Rosado (2015) indicaron que, el tiempo de respuesta en cuanto al inicio de la aplicación, la forma híbrida tardó 4.75 segundos en responder, mientras que su rival nativo tardó tan solo un segundo, como consiguiente dieron a conocer que la aplicación nativa es cuatro veces mejor que la otra (p. 85).
- **HE3:** Las aplicaciones nativas son más eficientes en cuanto a su conexión a la red. Lim (2015) indicó a través de resultados experimentales tanto para nativos como para aplicaciones híbridas. Afirmó que hay comunicación de datos sobrecargados para la aplicación híbrida, que existe una sobrecarga de comunicación de datos en la aplicación híbrida, que proviene de la capa del motor web de la aplicación híbrida. Todas las demás capacidades del dispositivo dan una sobrecarga similar en esta aplicación. Por lo tanto, la sobrecarga de esta aplicación existe en el estado actual de los sistemas móviles. A diferencia del híbrido el nativo tiene una mejor comunicación de datos y por lo general es muy estable (p. 10).
- **HE4:** Las aplicaciones nativas son mejores en cuanto a su carencia de seguridad. Gunawardhana (2021) indicó a través de pruebas de rendimiento que las aplicaciones nativas son mejores en cuanto a seguridad a comparación de las aplicaciones híbridas y las aplicaciones web. Sin embargo, en cuanto a velocidad de procesos, tanto las nativas como las híbridas obtienen buenos resultados en comparación de las aplicaciones web (p. 4647).

II. MARCO TEÓRICO

En el presente capítulo se manifiestan distintos estudios vinculados al proyecto de investigación, se investigaron y encontraron antecedentes tanto internacionales como nacionales, se detallaron teorías vinculadas al tema propuesto, asimismo de otras definiciones, Para obtener dicha información se realizó una exhaustiva búsqueda de artículos, libros, revistas, entre otras, en distintas bases de datos disponibles.

A continuación, se plasman investigaciones semejantes a la metodología para la evaluación del rendimiento de aplicaciones móviles, entre los cuales se privilegió las aplicaciones móviles nativas e híbridas, comparación y análisis.

Gunawardhana (2021) realizó una comparación para determinar qué método es mejor para desarrollar aplicaciones móviles, desarrolló tres aplicaciones con distintos métodos y luego evaluó estas aplicaciones que fueron una aplicación nativa, una aplicación híbrida y una aplicación web. Gunawardhana (2021) Dividió su evaluación en cuatro etapas y tomó características de evaluación la seguridad, usabilidad, estabilidad, uso de memoria, velocidad de procesos y duración de la batería. Gunawardhana (2021) llegó a la conclusión que una aplicación nativa es superior en portabilidad, una aplicación nativa brinda una mejor experiencia de usuario y que las aplicaciones web son menos amigables y menos efectivas que las anteriores.

Cárdenas et al. (2021) analizaron las tendencias de desarrollo de software para dispositivos móviles con el objetivo de compararlas a través del análisis y la bibliografía, en este estudio se incluye el uso de recursos, el nivel de interés y la usabilidad. La investigación que realizaron Cárdenas et al. (2021) fue de tipo analítica y bibliográfica, estableciendo indicadores de evaluación, además considera las búsquedas de los usuarios de internet entre los años 2015 y 2019 para evaluar el interés. Cárdenas et al. (2021) concluyeron que las aplicaciones con realidad aumentada y realidad virtual tienen un alto consumo de recursos, las aplicaciones con inteligencia artificial tienen un consumo medio de recursos, pero alta usabilidad y que los usuarios están más interesados en las aplicaciones basadas en la nube.

Paniagua, Bedoya y Mera (2020) desarrollaron un método para evaluar la usabilidad y accesibilidad en los aplicativos móviles incluyendo en su método los estándares de la WWC y las mejores prácticas de Web Móvil. Paniagua, Bedoya y Mera (2020) propusieron seis fases para evaluar la accesibilidad y usabilidad, establecieron como población objetiva a personas con discapacidad y recomendaron que la muestra sea al menos del 30% de los usuarios que utilizan la aplicación móvil, a dicha muestra se aplica una encuesta de satisfacción, con el reporte obtenido se procede a realizar un rediseño de la aplicación o una actualización para volver a ser sometida a una prueba. Paniagua, Bedoya y Mera (2020) llegaron a la conclusión que su método permite hacer una evaluación completa de la accesibilidad y usabilidad de una aplicación ya que en la última prueba llegaron a un 87% de opiniones positivas por parte de los usuarios. Finalmente recomendaron que es importante explicar a la población seleccionada la naturaleza de las pruebas para evitar deserciones durante el periodo de pruebas.

Granados (2020) describió métodos y enfoques para evaluar la usabilidad de una aplicación móvil. Granados (2020) realizó una revisión de la literatura, para analizar la experiencia de usuario al hacer uso de una aplicación móvil. Granados (2020) concluyó que una característica que afecta la aceptación en la usabilidad es la conectividad y que el uso de las aplicaciones móviles se incrementa rápidamente debido a la mejora de los dispositivos.

Dorfer, Demetz y Huber (2020) desarrollaron una investigación para determinar el impacto en el uso de recursos de una aplicación multiplataforma comparándola con una aplicación nativa para Android. Dorfer, Demetz y Huber (2020) ejecutaron una prueba que constaba de siete pasos para evaluar el consumo máximo y medio del CPU y la memoria, de la misma manera realizaron una prueba para evaluar el uso promedio y total de la batería. Dorfer, Demetz y Huber (2020) mostraron como resultado que las aplicaciones nativas utilizan mejor el CPU y la memoria, además que tienen menos consumo de la batería ya que indicaron que la aplicación multiplataforma desarrollada con React Native consumió entre un 6% y 8% más batería.

Lupo y Bellomo (2020) propusieron un marco metodológico para evaluar la calidad de un software desplegado en sistemas operativos móviles enfocándose más en la usabilidad. Lupo y Bellomo (2020) integraron DEMATEL junto con ANP para formar su modelo MCDA con el fin de evaluar la usabilidad del software, asimismo realizaron una evaluación comparativa con otras investigaciones para probar la validez de sus resultados obtenidos. Lupo y Bellomo (2020) demostraron la efectividad de su método y abrieron la posibilidad de combinar su método con otros para así mejorar el proceso de evaluación de un software.

Gomes y Silva (2019) realizaron un estudio comparativo entre dos herramientas de desarrollo móvil, Xamarin que permite la creación de una app mediante un solo lenguaje C# y CronApp que permite el desarrollo de una app híbrida. Gomes y Silva (2019) para dicha evaluación tomaron como criterio el rendimiento, característica, cantidad de datos y hora de lanzamiento de las dos herramientas de desarrollo móvil Xamarin y CronApp. Gomes y Silva (2019) concluyeron que Xamarin es una herramienta nativa muy eficaz de bajo costo que tiene como meta optimizar el tiempo de desarrollo. Esta solución se muestra como ventajosa por su fachada en el intercambio de los lenguajes en Android o iOS. Por consiguiente, Xamarin es una gran herramienta puesto que trabaja con instrumentos nativos.

Machini, Enriquez y Casas (2019) analizaron la calidad de las aplicaciones móviles haciendo uso de la librería Q2M. Machini, Enriquez y Casas (2019) mencionaron que al hacer uso de la librería utilizaron una métrica de valor numérico o nominal, si se obtiene datos positivos el valor de la métrica se incrementa y si ocurre lo contrario el valor se decrementa, tomaron como métricas la batería, el uso de memoria, uso del procesador, conexión a la red, etc. Machini, Enriquez y Casas (2019) llegaron a la conclusión que la librería Q2M permite hacer mediciones sobre la calidad de una aplicación móvil y que las métricas utilizadas para su evaluación pueden ser aplicadas fácilmente.

Usman, Iqbal y Khan (2020) propusieron un enfoque basado en modelos acoplados para la generación de pruebas de rendimiento en aplicaciones móviles. Usman, Iqbal y Khan (2020) identificaron cuatro variaciones diferentes

en una aplicación tales como los requerimientos funcionales, variaciones de hardware, variaciones de software y variaciones de sistemas operativos. El enfoque de prueba se centra en evaluar el rendimiento basado en el uso de recursos, en particular el tiempo de procesamiento, el uso de memoria y el consumo de batería. Usman, Iqbal y Khan (2020) llegaron a la conclusión que su enfoque propuesto tiene potencial para reducir significativamente el tiempo y el esfuerzo a la hora de realizar pruebas de rendimiento a las diferentes variantes de aplicaciones móviles.

Thomas et al. (2018) desarrollaron una investigación sobre las tendencias en el desarrollo móvil con el objetivo estudiar las principales características de ingeniería de software aplicadas en las variadas tendencias de desarrollo móvil comparando el rendimiento y la experiencia de usuario, analizando las ventajas y limitaciones de diferentes tipos de desarrollo móvil. Thomas et al. (2018) realizaron un estudio comparativo de los diferentes enfoques de desarrollo móvil, analizaron el rendimiento, las ventajas y desventajas de aplicaciones nativas, híbridas, web móviles, aplicaciones por compilación cruzada y aplicaciones móviles interpretadas. Thomas et al. (2018) expusieron como conclusiones las ventajas que tiene un aplicativo móvil nativo, el fácil acceso a los recursos del equipo y su alto rendimiento a diferencia de una aplicación híbrida que tiene una ejecución ralentizada, pero también resaltan la flexibilidad de una aplicación multiplataforma en las que incluye a aplicaciones híbridas y aplicaciones web progresivas.

Salgado (2018) realizó un estudio acerca de la eficiencia de apps móviles con un framework nativo e híbrido basado en Benchmarking, donde analizó herramientas que evalúen el tiempo de respuesta en un caso específico. Salgado (2018) empleó dos tipos de desarrollo específicos, los cuales fueron nativo e híbrido, a estas aplicaciones se les evaluó basándose en la norma ISO 25000 teniendo de enfoque principal el uso de recursos y el comportamiento de la aplicación. Salgado (2018) concluyó que el desarrollo nativo es más eficiente que el híbrido, sin embargo, hay ciertos ítems donde el desarrollo nativo recae ante el desarrollo híbrido, uno de ellos es en el espacio de almacenamiento y

consumo de internet donde llega a colapsar, cosa que en el método híbrido no pasa.

Venteu y Pinto (2018) enfocaron su trabajo sobre los tipos de formas de desarrollo móvil tanto híbrido como nativo, detallando cada uno de los procesos que conlleva al realizar dicha ejecución de desarrollo. Venteu y Pinto (2018) realizaron un enfoque detallado acerca de las tecnologías que conlleva el desarrollo híbrido, teniendo el lenguaje de marcado hipertexto (HTML), hoja de estilo cascada, el lenguaje de programación JavaScript, asimismo, mencionan Ionic como framework y para el entorno de desarrollo, PhoneGap y Cordova. Venteu y Pinto (2018) concluyeron que cuando se trata de rendimiento, hay una gran diferencia entre los dos tipos de desarrollo, teniendo como vencedor al nativo, sin embargo, los híbridos en cuanto a costo y tiempo de desarrollo son más eficaces.

Ahmad et al. (2018) identificaron los desafíos de las aplicaciones móviles nativas, híbridas y web exactamente los desafíos que puede afectar el desarrollo exitoso de dichas aplicaciones. Ahmad et al. (2018) desarrollaron su estudio basándose en otras investigaciones realizando una revisión sistemática de la literatura, identificando como principales desafíos la fragmentación, las pruebas, experiencia del usuario, reutilización de código, compatibilidad, falta de herramientas de soporte y seguridad. Ahmad et al. (2018) concluyeron que los estudios revisados presentan desafíos comunes, pero no se asignaron desafíos en mantenimiento de software, gestión de la configuración y el proceso de ingeniería de software.

Arias (2017) contrastó las diferencias que existen entre una aplicación híbrida y una aplicación web tomando como dimensión la usabilidad. Arias (2017) empleó la metodología DCU (Diseño Centrado en el Usuario), por consiguiente, comprendió el contexto de uso enfocándose en los requerimientos de los usuarios y construyó un prototipo de aplicación móvil híbrida donde se implementaron algunas mejoras obtenidas a través de una recopilación de información obtenidas por el CIEI (Centro de Investigación y Evaluación Institucional), por último, comparó los resultados y reconoció las diferencias que existen entre ambas aplicaciones. Arias (2017) llegó a la conclusión que es

mucho mejor desarrollar una aplicación, al margen de una ya existente, esto con el fin de mejorar el uso y experiencia del usuario mientras este use la aplicación.

Que, Guo y Zhu (2017) investigaron el paradigma de la aplicación híbrida y analizaron las ventajas y desventajas de ambos paradigmas en la teoría y experiencia. Que, Guo y Zhu (2017) realizaron una comparación teórica tomando como punto de vista la facilidad de codificación, depuración/prueba, distribución y actualización, facilidad de uso y sus funcionalidades, asimismo, realizaron una comparación de rendimiento tomando cinco indicadores. Que, Guo y Zhu (2017) concluyeron que, de acuerdo con los resultados del análisis y experimentos obtenidos, la aplicación híbrida se desarrolla y mantiene fácilmente, a cambio de eso debe reducir su rendimiento, a diferencia de las nativas dan una gran experiencia a los usuarios y un buen rendimiento.

Identificando diversas fuentes a continuación se desarrolla las teorías relacionadas al tema de investigación.

Una de las dimensiones identificadas para evaluar el rendimiento de una aplicación móvil es el tiempo de respuesta. Para Ma et al. (2018) es el intervalo desde el momento en el que se emite la primera solicitud de red hasta el momento en el que se recibe la última respuesta. El tiempo de respuesta no es equivalente al tiempo percibido por el usuario cuando usa una función. En su lugar, indica de forma exhaustiva cuánto tiempo puede realizarse a nivel del sistema (p. 4).

Otro factor importante en el rendimiento de las aplicaciones móviles es el uso de la batería. Para Ma et al. (2018) es la cantidad total de energía consumida desde el momento de la primera solicitud hasta su última respuesta. Además, recomienda aplicar el modelo energético V-Edge para calcular con precisión el consumo de energía (p. 5).

Un factor muy importante que puede afectar el rendimiento de una aplicación es el uso de recursos Dorfer, Demetz y Huber (2020) consideraron que el alto uso de CPU y memoria RAM afectan considerablemente a la batería (p. 196).

La conexión a la red influye demasiado en las aplicaciones móviles. Para Velásquez et al. (2019) las aplicaciones móviles cuya actividad requiera una conexión estable tienen la obligación de tomar en cuenta el tipo de red de transferencia, puesto que no tener en cuenta las propiedades de la infraestructura puede tener como consecuencia dificultades de rendimiento de la aplicación móvil (p. 43).

Por el lado de carencia de seguridad dentro de las aplicaciones móviles. Para Quiroz y Valencia (2017) las aplicaciones móviles cuya actividad requiera una conexión estable tienen la obligación de tomar en cuenta el tipo de red de transferencia, puesto que no tener en cuenta las propiedades de la infraestructura puede tener como consecuencia dificultades de rendimiento de la aplicación móvil (p. 43).

Identificando diversas fuentes a continuación se detallan conceptos importantes asociados a la presente investigación.

Para el presente estudio es importante conocer que es una aplicación móvil. Al respecto Angelova (2019) la definió como un software que está diseñado para ser instalado en un dispositivo móvil (p. 855).

Como ya se mencionó anteriormente existen varios tipos de aplicaciones móviles, Guo y Zhu (2017) indicaron que una aplicación híbrida está programada en un lenguaje compatible con el navegador y envuelta como aplicación nativa por eso se llama híbrida porque combina las características de ambas aplicaciones nativa y web. La aplicación híbrida se ve y funciona de forma nativa, pero en realidad se ejecuta como una aplicación web, además posee la gran experiencia de usuario de una aplicación nativa y es multiplataforma como una aplicación web (p. 611). Para Gunawardhana (2021) estas aplicaciones constan de complementos personalizables que se pueden acceder a las funciones nativas siempre que estén disponibles en los dispositivos, su desarrollo es 3 veces más rápido que las nativas y se puede ahorrar hasta el 80% de los costos en el desarrollo (p. 4644).

Por otra parte Gunawardhana (2021) señaló que una aplicación nativa tiene interfaces que están siempre conectadas con el sistema operativo y que puede

acceder a todos los servicios que este provee. Para desarrollar una aplicación nativa es necesario contar con diferentes habilidades ya que se requieren diferentes lenguajes según el sistema operativo del dispositivo (p. 4643). Además Léger et al. (2019) definieron a las aplicaciones nativas como software diseñado para ejecutarse solo en un sistema operativo móvil (SO) y, por lo tanto, no son multiplataforma, el beneficio de ser nativo es la capacidad de acceder y explotar todas las capacidades del dispositivo, las aplicaciones nativas son rápidas, receptivas y capaces de dar una experiencia óptima al usuario (p. 86).

III. METODOLOGÍA

3.1 Tipo y diseño de investigación

El tipo de investigación es aplicada porque se tiene como objetivo crear una metodología para evaluar el rendimiento de las aplicaciones móviles y aplicarla con el fin de determinar el rendimiento de una aplicación móvil nativa versus la misma aplicación desarrollada con tecnología híbrida teniendo en cuenta el uso de recursos, tiempo de respuesta, conexión a la red y carencia de seguridad. Al respecto Lozada (2014) afirmó que la investigación aplicada se centra en la producción de entendimiento con aplicación directa y a poco plazo en la población o en la zona productiva. Gracias a estos tipos de estudio se da una gran importancia por el uso del conocimiento que procede de la investigación básica (p. 35).

El enfoque de la investigación es cuantitativa ya que se busca medir el rendimiento de una aplicación móvil a través de una metodología, al respecto Hernández, Fernández y Baptista (2014) indicaron que este enfoque busca medir las variables de estudio con precisión, además se basa en investigaciones anteriores y que esta investigación se utiliza para consolidar hipótesis formuladas de forma lógica mediante un esquema teórico (p.10).

El diseño de investigación es descriptivo ya que se pretende recolectar y analizar datos sobre el rendimiento de una determinada aplicación móvil obtenidos al aplicar una metodología. Al respecto Hernández y Mendoza (2018) explicaron que este diseño tiene como objetivo especificar las características y propiedades de una variable en un determinado contexto (p. 108).

3.2 Variables y operacionalización

A continuación, se detalla en cada uno de los aspectos de la variable rendimiento de aplicaciones móviles:

3.2.1 Definición conceptual: “El rendimiento es el conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesarios bajo condiciones establecidas” (Norma ISO/IEC 9126). Además, Thomas et al. (2017) señalaron que debido a las variadas formas de crear aplicaciones móviles es necesario nuevas

metodologías y prácticas que beneficien el desarrollo de la ingeniería de software (p. 429).

3.2.2 Definición operacional: La variable se midió por medio de una ficha de observación, como dimensiones se consideran el Tiempo de respuesta, el uso de recursos, la conexión a la red y la carencia de seguridad.

3.2.3 Indicadores: Los indicadores considerados son el uso de CPU, uso de memoria RAM, uso de la batería, conexión a la red y carencia de seguridad.

3.2.4 Escala de medición: En la presente investigación se consideró la escala de medición de razón ya que se busca conocer el rendimiento de diferentes tipos de aplicaciones móviles.

3.3 Población, muestra y muestreo

A continuación, se detalla la población, muestra, muestreo y unidad de análisis de la presente investigación.

3.3.1 Población: La población de la presente investigación está conformada por las aplicaciones móviles nativas y aplicaciones móviles híbridas. Las que fueron evaluadas teniendo en cuenta el tiempo de respuesta, uso de CPU, uso de la memoria RAM, uso de la batería, conexión a la red y carencia de seguridad. Con respecto a la población Ventura (2017) la definió como un conjunto de elementos en los que se pretende estudiar ciertas características.

3.3.2 Muestra: En la muestra se consideró a dos aplicaciones móviles, una nativa y la misma pero desarrollada con tecnología híbrida. Al respecto Otzen y Manterola (2017) manifestaron que una muestra es un número que representa a la población permitiendo realizar inferencias y generar conclusiones.

3.3.3 Muestreo: El muestreo fue no probabilístico por conveniencia ya que para el presente estudio es necesario aplicar la metodología creada en aplicaciones que sean accesibles y que se conozca con qué tipo de desarrollo fue creada. Al respecto Otzen y Manterola (2017) señalaron que este tipo de muestreo permite elegir los casos que estén próximos y accesible a los investigadores.

3.3.4 Unidad de Análisis: Cada tipo de aplicaciones móviles que fue sometida a la medición.

3.4 Técnicas e instrumentos de recolección de datos

A continuación, se presentan los instrumentos de recolección de datos, su validez y su confiabilidad:

Se utilizó como técnica la observación, ya que se busca analizar el rendimiento de una aplicación, al respecto Hernández y Mendoza (2018) definieron a la observación como un método de recolección de datos válido y confiable útil para analizar el comportamiento (p. 290) y como instrumento de recolección de datos se utilizó la guía de observación para recolectar los datos obtenidos luego de aplicar la metodología ver (Anexo 7 al 13).

Se utilizó la validez de contenido, puesto que se desarrolla una metodología basada en los antecedentes de los que se tiene en cuenta las mejores prácticas, técnicas, métodos e instrumentos que utilizan para evaluar el rendimiento de una aplicación móvil, al respecto Hernández y Mendoza (2018) indicaron que la validez es el grado en que un instrumento pretende medir la variable y que en la validez de contenido el instrumento refleja un dominio específico de la variable medida (p. 229-230).

Con respecto a la confiabilidad, no realizó una medición ya que el instrumento utilizado fue la ficha de recolección de datos. Por lo tanto, se consideró el 95% de nivel de confianza en la aplicación de las pruebas estadísticas. Ñaupá et al. (2018) indicaron que un instrumento es confiable si al volver a aplicar la misma prueba, los resultados no varían significativamente sin importar el tiempo (p. 277).

3.5 Procedimientos

En la presente investigación se elaboró una metodología para evaluar el rendimiento de aplicaciones móviles la cual cuenta con las siguientes fases: Planeación, implementación y pruebas y análisis de datos.

En la sección anexos se encuentra la estructura detallada de la metodología la cual cuenta con los siguientes procesos:

- a) Seleccionar las aplicaciones móviles (Proceso 01 de la metodología MERAM).
- b) Preparar el ambiente de pruebas (Proceso 02 de la metodología MERAM).
- c) Realizar pruebas de uso de recursos (Proceso 03 de la metodología MERAM).
- d) Realizar pruebas de tiempo de respuesta (Proceso 04 de la metodología MERAM).
- e) Realizar pruebas de conexión a la red (Proceso 05 de la metodología MERAM).
- f) Realizar pruebas de carencia de seguridad (Proceso 06 de la metodología MERAM).

3.6 Método de análisis de datos

En esta sección se describe el método de análisis de datos adecuado al presente proyecto de investigación.

Para realizar la prueba de normalidad se utilizó Shapiro-Wilks y Kolmogorov-Smirnov con el fin de verificar si los datos tabulados tienen o no una distribución normal. Al respecto Romero (2016) señaló que para contrastar la normalidad de una muestra menor o igual a 50 se aplica Shapiro-Wilks (p.43). Además, Kim y Park (2019) señalaron la prueba de Shapiro-Wilks, se conoce como una de las pruebas de normalidad más poderosas, teóricamente es posible realizar la prueba de normalidad con tres muestras. Sin embargo, incluso si el valor P es mayor que el nivel de significancia de 0.05, esto no significa automáticamente que los datos sigan una distribución normal (p. 02). Si la significancia es mayor a 0.05 significa que la distribución es normal, si la significancia es menor a 0.05 se procedería a utilizar la prueba de Wilcoxon. Sánchez (2015) indicó que la diferencia de dispersión de datos entre un grupo y otro es la esencia de esta prueba (p. 21).

3.7 Aspectos éticos

Para el presente estudio se tomó en consideración las referencias bibliográficas de diversos autores, libros, artículos de investigación y tesis. Además, el informe de investigación fue desarrollado siguiendo las normas y los lineamientos del Vicerrectorado de Investigación de la Universidad César Vallejo.

Conforme al código de ética del colegio de ingenieros del Perú se ejecuta correctamente en el presente informe de investigación referente a los artículos como, Artículo 13° señala con respecto al comportamiento profesional del ingeniero y su conducta tiene la obligación de ser establecidos con los objetivos de la institución, asimismo, con respecto a las conductas establecidas en el Artículo 15° las personas o miembros de la investigación se basan en la lealtad profesional, respeto, inclusión social, la honestidad, justicia, solidaridad y el honor profesional, los cuales tiene que contar dentro de la investigación (Colegio de Ingenieros, 1999, p.2-3).

El código de ética de la Universidad César Vallejo se ocupa de custodiar por las investigaciones, que ejerzan los altos patrones de dureza científica, honestidad, responsabilidad e integridad de información recopilada resguardando los derechos y bienestar de los investigadores (Universidad César Vallejo, 2017).

Para realizar una investigación existe un conjunto de normas que decretan los mejores ensayos, garantizando y certificando a través de los fundamentos éticos la autoría de estudios que colaboran a otras investigaciones. Por lo tanto, esta investigación es verídica con respecto a los aspectos éticos que solicita la Universidad Cesar Vallejo en todos los proyectos de investigación.

IV. RESULTADOS

En el presente capítulo se describen los resultados adquiridos de la investigación en base a los indicadores de rendimiento de las aplicaciones móviles nativa e híbrida, tales como uso de CPU, uso de la memoria RAM, uso de la batería, tiempo de respuesta, conexión a la red y carencia de seguridad. Igualmente, se muestra el procedimiento de los datos adquiridos al aplicar la metodología. Los datos recolectados fueron procesados mediante el software SPSS.

A continuación, se muestran los datos descriptivos y la prueba de normalidad.

Los resultados obtenidos para el indicador uso del CPU fueron el resultado del procesamiento de los datos registrados de los parámetros de rendimiento instalación de la aplicación, inicio de la aplicación, ejecución de la aplicación y ejecución de la aplicación en segundo plano.

Tabla 1: Estadísticos descriptivos indicador uso del CPU

Descriptivos				
			Estadístico	Error típ.
Nativa	Media		20.59	1.419
	Intervalo de confianza para la media al 95%	Límite inferior	17.76	
		Límite superior	23.41	
	Media recortada al 5%		20.79	
	Mediana		24.00	
	Varianza		161.157	
	Desv. típ.		12.695	
	Mínimo		0	
	Máximo		38	
	Rango		38	
	Amplitud intercuartil		25	
	Asimetría		-.778	.269
Curtosis		-.857	.532	
Híbrida	Media		24.04	1.626
	Intervalo de confianza para la media al 95%	Límite inferior	20.80	
		Límite superior	27.27	
	Media recortada al 5%		24.40	
	Mediana		30.00	
	Varianza		211.530	
	Desv. típ.		14.544	

	Mínimo	0	
	Máximo	44	
	Rango	44	
	Amplitud intercuartil	29	
	Asimetría	-.903	.269
	Curtosis	-.811	.532

En los resultados descriptivos obtenidos para el indicador uso del CPU la aplicación nativa registra como una media de 20.59 y la aplicación híbrida 24.04, siendo menor el uso del CPU en la aplicación móvil nativa.

Para el indicador uso del CPU se aplicó la prueba de Kolmogorov Smirnov ya que el tamaño de la muestra es mayor a 50.

Tabla 2: Prueba de normalidad indicador uso del CPU

Pruebas de normalidad			
Aplicación móvil	Kolmogorov-Smirnov		
	Estadístico	gl	Sig.
Nativa	.238	80	.000
Híbrida	.256	80	.000

En los resultados de la prueba de normalidad para el indicador uso del CPU los valores significancia son menor a 0.05, esto quiere decir que los datos no siguen una distribución normal.

Los resultados obtenidos para el indicador uso de la memoria RAM fueron el resultado del procesamiento de los datos registrados de los parámetros de rendimiento instalación de la aplicación, inicio de la aplicación, ejecución de la aplicación y ejecución de la aplicación en segundo plano.

Tabla 3: Estadísticos descriptivos indicador uso de la memoria RAM

Descriptivos			
		Estadístico	Error típ.
Nativa	Media	13.96	.705
	Intervalo de confianza para la media al 95%	Límite inferior	12.56
		Límite superior	15.36
	Media recortada al 5%	13.93	

	Mediana	15.00	
	Varianza	39.707	
	Desv. típ.	6.301	
	Mínimo	3	
	Máximo	26	
	Rango	23	
	Amplitud intercuartil	10	
	Asimetría	-.350	.269
	Curtosis	-.738	.532
Híbrida	Media	15.09	.760
	Intervalo de confianza para la media al 95%	Límite inferior	13.58
		Límite superior	16.60
	Media recortada al 5%	15.11	
	Mediana	16.00	
	Varianza	46.182	
	Desv. típ.	6.796	
	Mínimo	4	
	Máximo	27	
	Rango	23	
	Amplitud intercuartil	11	
	Asimetría	-.441	.269
	Curtosis	-.820	.532

En los resultados descriptivos obtenidos para el indicador uso de memoria RAM, la aplicación nativa registra como una media de 13.96 y la aplicación híbrida 15.09, siendo menor el uso de la memoria RAM en la aplicación móvil nativa.

Para el indicador uso de la memoria RAM se aplicó la prueba de Kolmogorov Smirnov ya que el tamaño de la muestra es mayor a 50.

Tabla 4: Prueba de normalidad indicador uso de la memoria RAM

Pruebas de normalidad			
Aplicación móvil	Kolmogorov-Smirnov		
	Estadístico	gl	Sig.
Nativa	.240	80	.000
Híbrida	.232	80	.000

En los resultados de la prueba de normalidad para el indicador uso de memoria RAM los valores significancia son menor a 0.05, esto quiere decir que los datos no siguen una distribución normal.

Los resultados obtenidos para el indicador uso de la batería fueron el resultado del procesamiento de los datos registrados de los parámetros de rendimiento uso de la aplicación durante (10, 20, 30 y 60) minutos y uso de la aplicación en segundo plano durante (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 y 24) horas.

Tabla 5: Estadísticos descriptivos indicador uso de la batería

Descriptivos				
			Estadístico	Error típ.
Nativa	Media		2.71	.803
	Intervalo de confianza para la media al 95%	Límite inferior	1.00	
		Límite superior	4.41	
	Media recortada al 5%		2.23	
	Mediana		2.00	
	Varianza		10.971	
	Desv. típ.		3.312	
	Mínimo		0	
	Máximo		14	
	Rango		14	
	Amplitud intercuartil		2	
	Asimetría		2.773	.550
Curtosis		8.879	1.063	
Hibrida	Media		3.41	.970
	Intervalo de confianza para la media al 95%	Límite inferior	1.35	
		Límite superior	5.47	
	Media recortada al 5%		2.90	
	Mediana		2.00	
	Varianza		16.007	
	Desv. típ.		4.001	
	Mínimo		0	
	Máximo		16	
	Rango		16	
	Amplitud intercuartil		3	
	Asimetría		2.291	.550
Curtosis		5.822	1.063	

En los resultados descriptivos obtenidos para el indicador uso de la batería, la aplicación nativa registra como una media de 2.71 y la aplicación híbrida 3.41, siendo menor el uso de la batería en la aplicación móvil nativa.

Para el indicador uso de la batería se aplicó la prueba de Shapiro Wilk ya que el tamaño de la muestra es menor a 50.

Tabla 6: Prueba de normalidad indicador uso de la batería

Pruebas de normalidad			
Aplicación móvil	Shapiro-Wilk		
	Estadístico	gl	Sig.
Nativa	.667	17	.000
Híbrida	.727	17	.000

En los resultados de la prueba de normalidad para el indicador uso de la batería los valores significancia son menor a 0.05, esto quiere decir que los datos no siguen una distribución normal.

Los resultados obtenidos para el indicador tiempo de respuesta fueron el resultado del procesamiento de los datos registrados de los parámetros de rendimiento instalación de la aplicación, inicio de la aplicación, y ejecución de la aplicación.

Tabla 7: Estadísticos descriptivos indicador tiempo de respuesta

Descriptivos				
		Estadístico	Error típ.	
Nativa	Media	4808.98	369.906	
	Intervalo de confianza para la media al 95%	Límite inferior	4068.80	
		Límite superior	5549.16	
	Media recortada al 5%	4774.63		
	Mediana	4412.50		
	Varianza	8209827.983		
	Desv. típ.	2865.280		
	Mínimo	1421		
	Máximo	8922		
	Rango	7501		
Amplitud intercuartil	6783			

	Asimetría	.222	.309	
	Curtosis	-1.513	.608	
Híbrida	Media	5215.20	408.517	
	Intervalo de confianza para la media al 95%	Límite inferior	4397.76	
		Límite superior	6032.64	
	Media recortada al 5%	5178.74		
	Mediana	4745.50		
	Varianza	10013177.892		
	Desv. típ.	3164.361		
	Mínimo	1501		
	Máximo	9684		
	Rango	8183		
	Amplitud intercuartil	7444		
	Asimetría	.219	.309	
	Curtosis	-1.526	.608	

En los resultados descriptivos obtenidos para el indicador tiempo de respuesta, la aplicación nativa registra como una media de 4808.98 y la aplicación híbrida 5215.20, siendo menor el tiempo respuesta en la aplicación móvil nativa.

Para el indicador tiempo de respuesta se aplicó la prueba de Kolmogorov Smirnov ya que el tamaño de la muestra es mayor a 50.

Tabla 8: Prueba de normalidad indicador tiempo de respuesta

Pruebas de normalidad			
Aplicación móvil	Kolmogorov-Smirnov		
	Estadístico	gl	Sig.
Nativa	.206	60	.000
Híbrida	.215	60	.000

En los resultados de la prueba de normalidad para el indicador tiempo de respuesta los valores significancia son menor a 0.05, esto quiere decir que los datos no siguen una distribución normal.

Los resultados obtenidos para el indicador conexión a la red fueron el resultado del procesamiento de los datos registrados de los parámetros de rendimiento inicio de la aplicación, carga de archivos de (512 KB, 1 MB, 5 MB y 10 MB), descarga de archivos de (512 KB, 1 MB, 5 MB y 10 MB).

Tabla 9: Estadísticos descriptivos indicador conexión a la red

Descriptivos				
			Estadístico	Error típ.
Nativa	Media		3326.75	136.806
	Intervalo de confianza para la media al 95%	Límite inferior	3056.97	
		Límite superior	3596.53	
	Media recortada al 5%		3208.52	
	Mediana		2805.50	
	Varianza		3743170.731	
	Desv. típ.		1934.728	
	Mínimo		866	
	Máximo		8168	
	Rango		7302	
	Amplitud intercuartil		2688	
	Asimetría		.908	.172
	Curtosis		.226	.342
Híbrida	Media		3473.53	141.699
	Intervalo de confianza para la media al 95%	Límite inferior	3194.11	
		Límite superior	3752.95	
	Media recortada al 5%		3366.20	
	Mediana		2918.00	
	Varianza		4015733.547	
	Desv. típ.		2003.930	
	Mínimo		785	
	Máximo		8160	
	Rango		7375	
	Amplitud intercuartil		2832	
	Asimetría		.809	.172
	Curtosis		-.097	.342

En los resultados descriptivos obtenidos para el indicador conexión a la red, la aplicación nativa registra como una media de 3326.75 y la aplicación híbrida 3473.53, siendo menor el tiempo de transferencia de archivos en la aplicación móvil nativa.

Para el indicador conexión a la red se aplicó la prueba de Kolmogorov Smirnov ya que el tamaño de la muestra es mayor a 50.

Tabla 10: Prueba de normalidad indicador conexión a la red

Pruebas de normalidad			
Aplicación móvil	Kolmogorov-Smirnov		
	Estadístico	gl	Sig.
Nativa	.138	200	.000
Hibrida	.135	200	.000

En los resultados de la prueba de normalidad para el indicador conexión a la red los valores significancia son menor a 0.05, esto quiere decir que los datos no siguen una distribución normal.

Los resultados obtenidos para el indicador carencia de seguridad fueron el resultado del procesamiento de los datos registrados de los parámetros de rendimiento configuraciones incorrectas, secuestro de tareas, seguridad de red, suplantación de identidad, respaldo, uso de componentes y permisos innecesarios.

Tabla 11: Estadísticos descriptivos indicador carencia de seguridad

Descriptivos				
		Estadístico	Error típ.	
Nativa	Media	1.43	.272	
	Intervalo de confianza para la media al 95%	Límite inferior	.84	
		Límite superior	2.02	
	Media recortada al 5%	1.42		
	Mediana	1.00		
	Varianza	1.033		
	Desv. típ.	1.016		
	Mínimo	0		
	Máximo	3		
	Rango	3		
	Amplitud intercuartil	1		
	Asimetría	.481	.597	
Curtosis	-.736	1.154		
Hibrida	Media	1.21	.350	
	Intervalo de confianza para la media al 95%	Límite inferior	.46	
		Límite superior	1.97	
	Media recortada al 5%	1.07		

Mediana	1.00	
Varianza	1.720	
Desv. típ.	1.311	
Mínimo	0	
Máximo	5	
Rango	5	
Amplitud intercuartil	2	
Asimetría	1.929	.597
Curtosis	5.065	1.154

En los resultados descriptivos obtenidos para el indicador carencia de seguridad, la aplicación nativa registra como una media de 1.43 y la aplicación híbrida 1.21, siendo menor la carencia de seguridad en aplicación móvil nativa.

Para el indicador carencia de seguridad se aplicó la prueba de Shapiro Wilk ya que el tamaño de la muestra es menor a 50.

Tabla 12: Prueba de normalidad indicador carencia de seguridad

Pruebas de normalidad			
Aplicación móvil	Shapiro-Wilk		
	Estadístico	gl	Sig.
Nativa	.844	14	.018
Híbrida	.767	14	.002

En los resultados de la prueba de normalidad para el indicador carencia de seguridad los valores significancia son menor a 0.05, esto quiere decir que los datos no siguen una distribución normal.

A continuación, se muestra el resultado de las pruebas de hipótesis.

HE1o: Las aplicaciones nativas no son más eficientes en cuanto a uso de recursos.

HE1a: Las aplicaciones nativas son más eficientes en cuanto a uso de recursos.

Debido a que las pruebas de normalidad en el indicador de uso del CPU no siguen una distribución normal se procedió a realizar la prueba de Wilcoxon.

Tabla 13: Prueba de Wilcoxon indicador uso del CPU

Rangos				
		N	Rango promedio	Suma de rangos
Uso del CPU Híbrida – Uso del CPU Nativa	Rangos negativos	13 ^a	11.62	151.00
	Rangos positivos	44 ^b	34.14	1502.00
	Empates	23 ^c		
	Total	80		

Tabla 14: Prueba de Wilcoxon indicador uso de la memoria RAM

Rangos				
		N	Rango promedio	Suma de rangos
Uso de memoria RAM Híbrida – Uso de memoria RAM Nativa	Rangos negativos	10 ^a	19.95	199.50
	Rangos positivos	48 ^b	31.49	1511.50
	Empates	22 ^c		
	Total	80		

Tabla 15: Prueba de Wilcoxon indicador uso de la batería

		N	Rango promedio	Suma de rangos
Uso de la batería Híbrida – Uso de la batería Nativa	Rangos negativos	0 ^a	0.00	0.00
	Rangos positivos	8 ^b	4.50	36.00
	Empates	9 ^c		
	Total	17		

En las siguientes tablas se muestran los estadísticos de prueba para la hipótesis específica 1 donde se muestra el valor de significancia.

Tabla 16: Prueba de hipótesis – Uso de recursos (Uso del CPU)

Estadísticos de contraste ^a	
	Híbrida - Nativa
Z	-5,380 ^b
Sig. asintót. (bilateral)	.000

Tabla 17: Prueba de hipótesis – Uso de recursos (Uso de la memoria RAM)

Estadísticos de contraste ^a	
	Híbrida - Nativa
Z	-5,153 ^b

Sig. asintót. (bilateral)	.000
---------------------------	------

Tabla 18: Prueba de hipótesis – Uso de recursos (Uso de la batería)

Estadísticos de contraste ^a	
	Híbrida - Nativa
Z	-2,588 ^b
Sig. asintót. (bilateral)	.010

Según los resultados obtenidos de la prueba de Wilcoxon, el nivel de significancia fue menor que 0.05. Por lo que se rechaza la hipótesis nula y se acepta la hipótesis alternativa, por lo tanto, se puede afirmar que una aplicación nativa es más eficiente en cuanto al uso de recursos que una aplicación híbrida.

HE2o: Las aplicaciones nativas no son más eficientes en cuanto al tiempo de respuesta.

HE2a: Las aplicaciones nativas son más eficientes en cuanto al tiempo de respuesta.

Debido a que las pruebas de normalidad en el indicador de tiempo de respuesta no siguen una distribución normal se procedió a realizar la prueba de Wilcoxon.

Tabla 19: Prueba de Wilcoxon indicador tiempo de respuesta

		N	Rango promedio	Suma de rangos
Tiempo de respuesta Híbrida – Tiempo de respuesta Nativa	Rangos negativos	5 ^a	18.00	90.00
	Rangos positivos	55 ^b	31.64	1740.00
	Empates	0 ^c		
	Total	60		

En la siguiente tabla se muestran los estadísticos de prueba para la hipótesis específica 2 donde se muestra el valor de significancia.

Tabla 20: Prueba de hipótesis – Tiempo de respuesta

Estadísticos de contraste ^a	
	Tiempo de respuesta Hibrida – Tiempo de respuesta Nativa
Z	-6,073 ^b
Sig. asintót. (bilateral)	.000

Según los resultados obtenidos de la prueba de Wilcoxon, el nivel de significancia fue de 0.00 siendo menor que 0.05. Por lo que se rechaza la hipótesis nula y se acepta la hipótesis alternativa, por lo tanto, se puede afirmar que una aplicación nativa es más eficiente en cuanto al tiempo de respuesta que una aplicación híbrida.

HE3o: Las aplicaciones nativas no son más eficientes en cuanto a su conexión a la red.

HE3a: Las aplicaciones nativas son más eficientes en cuanto a su conexión a la red.

Debido a que las pruebas de normalidad en el indicador conexión a la red no siguen una distribución normal se procedió a realizar la prueba de Wilcoxon.

Tabla 21: Prueba de Wilcoxon indicador conexión a la red

		N	Rango promedio	Suma de rangos
Tiempo de respuesta Hibrida – Tiempo de respuesta Nativa	Rangos negativos	51 ^a	82.22	4193.00
	Rangos positivos	147 ^b	105.50	15508.00
	Empates	2 ^c		
	Total	200		

En la siguiente tabla se muestran los estadísticos de prueba para la hipótesis específica 3 donde se muestra el valor de significancia.

Tabla 22: Prueba de hipótesis – Conexión a la red

Estadísticos de contraste ^a	
	Tiempo de respuesta Hibrida – Tiempo de respuesta Nativa

Z	-7,008 ^b
Sig. asintót. (bilateral)	.000

Según los resultados obtenidos de la prueba de Wilcoxon, el nivel de significancia fue de 0.00 siendo menor que 0.05. Por lo que se rechaza la hipótesis nula y se acepta la hipótesis alternativa, por lo tanto, se puede afirmar que una aplicación nativa es más eficiente en cuanto a la conexión a la red que una aplicación híbrida.

HE4o: Las aplicaciones nativas no son mejores en cuanto a su carencia de seguridad.

HE4a: Las aplicaciones nativas son mejores en cuanto a su carencia de seguridad.

Debido a que las pruebas de normalidad en el indicador de carencia de seguridad no siguen una distribución normal se procedió a realizar la prueba de Wilcoxon.

Tabla 23: Prueba de Wilcoxon indicador carencia de seguridad

Rangos				
		N	Rango promedio	Suma de rangos
Carencia de seguridad - Híbrido – Carencia de seguridad Nativa	Rangos negativos	4 ^a	3.25	13.00
	Rangos positivos	2 ^b	4.00	8.00
	Empates	8 ^c		
	Total	14		

En la siguiente tabla se muestran los estadísticos de prueba para la hipótesis específica 4 donde se muestra el valor de significancia.

Tabla 24: Prueba de hipótesis – Carencia de seguridad

Carencia de seguridad Híbrida – Carencia de seguridad Nativa	
Z	-,530 ^b
Sig. asintót. (bilateral)	.596

Según los resultados obtenidos de la prueba de Wilcoxon, el nivel de significancia fue de 0.596 siendo mayor que 0.05. Por lo que se rechaza la hipótesis alternativa, y no se puede afirmar que la aplicación nativa es mejor en cuanto a la carencia de seguridad que una aplicación híbrida.

En la presente investigación se plantearon 4 hipótesis específicas, a continuación, se muestra el resumen de la aceptación o rechazo para cada una de las hipótesis planteadas.

Tabla 25: Resumen de las pruebas de hipótesis

Hipótesis específicas	Resultados	Condición
<p>HE1o: Las aplicaciones nativas no son más eficientes en cuanto a uso de recursos.</p> <p>HE1a: Las aplicaciones nativas son más eficientes en cuanto a uso de recursos.</p>	Según los resultados obtenidos de la prueba de Wilcoxon, el nivel de significancia fue menor que 0.05. Por lo que se rechaza la hipótesis nula y se acepta la hipótesis alternativa, por lo tanto, se puede afirmar que una aplicación nativa es más eficiente en cuanto al uso de recursos que una aplicación híbrida.	Aceptada
<p>HE2o: Las aplicaciones nativas no son más eficientes en cuanto al tiempo de respuesta.</p> <p>HE2a: Las aplicaciones nativas son más eficientes en cuanto al tiempo de respuesta.</p>	Según los resultados obtenidos de la prueba de Wilcoxon, el nivel de significancia fue de 0.00 siendo menor que 0.05. Por lo que se rechaza la hipótesis nula y se acepta la hipótesis alternativa, por lo tanto, se puede afirmar que una aplicación nativa es más eficiente en cuanto al tiempo de respuesta que una aplicación híbrida.	Aceptada
<p>HE3o: Las aplicaciones nativas no son más eficientes en cuanto a su conexión a la red.</p> <p>HE3a: Las aplicaciones nativas son más eficientes en cuanto a su conexión a la red.</p>	Según los resultados obtenidos de la prueba de Wilcoxon, el nivel de significancia fue de 0.00 siendo menor que 0.05. Por lo que se rechaza la hipótesis nula y se acepta la hipótesis alternativa, por lo tanto, se puede afirmar que una aplicación nativa es más eficiente en cuanto a la conexión a la red que una aplicación híbrida.	Aceptada
<p>HE4o: Las aplicaciones nativas no son mejores en cuanto a su carencia de</p>	Según los resultados obtenidos de la prueba de Wilcoxon, el nivel de significancia fue de 0.596 siendo mayor que 0.05. Por lo que se	Rechazada

seguridad. HE4a: Las aplicaciones nativas son mejores en cuanto a su carencia de seguridad.	rechaza la hipótesis alternativa, y no se puede afirmar que la aplicación nativa es mejor en cuanto a la carencia de seguridad que una aplicación híbrida.	
---	--	--

V. DISCUSIÓN

En esta sección se desarrolla la discusión del proyecto de investigación luego de haber sustraído el análisis de los resultados de cada indicador con el apoyo del software SPSS.

Con respecto a la hipótesis 1, las aplicaciones nativas son más eficientes en cuanto al uso de recursos. Se extrajo resultados para los indicadores uso de CPU, uso de memoria RAM y uso de la batería, para el indicador uso de CPU, la aplicación nativa registra una media de 20.59% y la aplicación híbrida 24.04%, siendo menor el uso del CPU en la aplicación móvil nativa, los resultados fueron similares al estudio de Durán, Peinado y Rosado (2015) ellos indicaron que en cuanto a la utilización del CPU la aplicación híbrida está por arriba del 50% y una la aplicación nativa emplea únicamente 2,5% de CPU, una diferencia abismal de parte de esta (p. 86). Sin embargo, está medición solo se da en el inicio de la aplicación y también se puede identificar que con el avance de la tecnología híbrida esta brecha se viene reduciendo.

Con respecto a la hipótesis 2, las aplicaciones nativas son más eficientes en cuanto al tiempo de respuesta. Se extrajo resultados para el indicador tiempo de respuesta, donde la aplicación nativa registra una media de 4808.98 milisegundos y la aplicación híbrida 5215.20 milisegundos, siendo menor el tiempo de respuesta en la aplicación móvil nativa, esto concuerda con los resultados obtenidos por Durán, Peinado y Rosado (2015) ellos indicaron que, el tiempo de respuesta en cuanto al inicio de la aplicación, la forma híbrida tardó 4,75 segundos en responder, mientras que su rival nativo tardó tan solo 1 segundo, como consiguiente dieron a conocer que la aplicación nativa es cuatro veces mejor que la otra.

Con respecto a la hipótesis 3, las aplicaciones nativas son más eficientes en cuanto a su conexión a la red. Se extrajo resultados para el indicador conexión a la red, donde la aplicación nativa registra una media de 3326.75 milisegundos y la aplicación híbrida 3473.53 milisegundos, siendo más rápida la transferencia de archivos en la aplicación nativa, esto concuerda con los resultados obtenidos por Lim (2015) quien afirmó que hay comunicación de datos sobrecargados para la aplicación híbrida, que existe una sobrecarga de comunicación de datos en la aplicación híbrida, que proviene de la capa del

motor web de la aplicación híbrida, a diferencia del híbrido el nativo tiene una mejor comunicación de datos y por lo general es muy estable. Sin embargo, se puede observar que no existe una gran diferencia entre las dos aplicaciones.

Con respecto a la hipótesis 4, las aplicaciones nativas son mejores en cuanto a su carencia de seguridad. Se extrajo resultados del indicador carencia de seguridad, donde la aplicación nativa registra un nivel de significancia de 0.596 por lo que no se puede afirmar que la aplicación nativa es mejor que la aplicación híbrida. Sin embargo, Gunawardhana (2021) indicó que las aplicaciones nativas son mejores en cuanto a seguridad a comparación de las aplicaciones híbridas y las aplicaciones web. pero en cuanto a velocidad de procesos, tanto las nativas como las híbridas obtienen buenos resultados.

VI. CONCLUSIONES

Empleando la metodología elaborada se determinó que las aplicaciones móviles nativas son más eficientes en cuanto al uso de recursos que las aplicaciones móviles híbridas ya que se obtuvo que la aplicación nativa utiliza 3.45% menos CPU, 1.126% menos memoria RAM y 0.71% menos batería que su similar híbrida.

De la misma manera, empleando la metodología se determinó que las aplicaciones móviles nativas son más eficientes en cuanto al tiempo de respuesta que las aplicaciones móviles híbridas ya que el tiempo de respuesta en la aplicación nativa es menor registrando una media de 4808 milisegundos y la aplicación híbrida 5215 milisegundos.

A la vez también empleando la metodología se determinó que las aplicaciones móviles nativas son más eficientes en cuanto a su conexión a la red que las aplicaciones móviles híbridas, esto se comprueba midiendo el tiempo de inicio de sesión, carga y descarga de archivos de (512 KB, 1 MB, 5 MB, 10 MB y 20MB) donde la aplicación nativa registra una media de 3326 milisegundos y la aplicación híbrida 3473 milisegundos.

Por último, empleando la metodología no se puede afirmar que las aplicaciones móviles nativas son mejores en cuanto a la carencia de seguridad que las aplicaciones móviles híbridas debido a que se obtuvo un nivel de significancia de 0.596.

VII. RECOMENDACIONES

Se sugiere que para evaluar el uso de recursos de las aplicaciones móviles tanto híbridas como nativas, esta medición se realice en el mismo dispositivo o en dispositivos con las mismas características, además que las aplicaciones móviles tengan funcionalidades similares ya que teniendo esto en cuenta se conocerá cual es la más eficiente.

Se recomienda ampliar la medición de las funcionalidades de acuerdo a las características de las aplicaciones a evaluar para tener un resultado más exacto del tiempo de respuesta.

Se recomienda evaluar la conexión a la red en las aplicaciones móviles teniendo una conexión que mantenga una velocidad constante de carga y descarga, de preferencia conexión a fibra óptica.

Se recomienda realizar los cambios sugeridos luego de aplicar la metodología para mejorar la carencia de seguridad en las aplicaciones móviles

Por último, se recomienda ampliar la medición de usabilidad incorporando la calificación de los usuarios finales de las aplicaciones.

REFERENCIAS

- AHMAD, A., LI, K., FENG, C., ASIM, S.M., YOUSIF, A. y GE, S., An Empirical Study of Investigating Mobile Applications Development Challenges. *IEEE Access*, 2018. vol. 6, pp. 17711-17728. ISSN 2169-3536.
- ANGELOVA, N., Mobile Applications for Business. *Trakia Journal of Sciences*, 2019. vol. 17, pp. 853-859. ISSN 1312-7069.
- ARIAS CHING, P., *Comparación de la usabilidad de una aplicación web y una aplicación híbrida en dispositivos*. S.l.: Universidad de Costa Rica. 2017.
- BOSNIC, S., PAPP, I. y NOVAK, S., 2016. The development of hybrid mobile applications with Apache Cordova. *24th Telecommunications Forum, TELFOR 2016*. S.l.: s.n., ISBN 9788674666494.
- CÁRDENAS VILLAVICENCIO, O.E., ZEA ORDÓÑEZ, M.P., VALAREZO PARDO, M.R. y RAMÓN RAMÓN, R.A., Comparativa de tendencias de desarrollo de software móvil. *3C TIC: Cuadernos de desarrollo aplicados a las TIC*, 2021. vol. 10, no. 1, pp. 123-147. ISSN 2254-6529.
- CHAVERRI CHAVES, D., Delimitation and Justification on Social Sciences research questions. *Revista ciencias sociales*, 2017. vol. 157, no. 3, pp. 185-193. ISSN 0482-5276.
- DEL PERÚ, Colegio de Ingenieros. Código de Ética del CIP. Código de ética del CIP, 1999, vol. 26.
- DELÍA, L.N., *Desarrollo De Aplicaciones Móviles Multiplataforma*. S.l.: Universidad Nacional de La Plata. 2017.
- DÍAZ, D., HERRERA, S. y ROSENZVAIG, F., 2019. Estudio de la Capacidad de Mantenimiento de las Aplicaciones Móviles Híbridas. *XXV Congreso Argentino de Ciencias de la Computacion*. S.l.: s.n., pp. 662-673. ISBN 978-987-688-377-1.
- DORFER, T., DEMETZ, L. y HUBER, S., Impact of mobile cross-platform development on CPU, memory and battery of mobile devices when using common mobile app features. *Procedia Computer Science*, 2020. vol. 175, no. 2019, pp. 189-196. ISSN 1877-0509.
- DURÁN SANJUÁN, A.J., PEINADO RODRÍGUEZ, J.L. y ROSADO, A.A.,

Comparación De Dos Tecnologías De Desarrollo De Aplicaciones Móviles Desde La Perspectiva De Los Atributos De Calidad. *Scientia Et Technica*, 2015. vol. 20, no. 1, pp. 81-87. ISSN 0122-1701.

GOMES BARBOSA, F.F. y SILVA DOS SANTOS, T., Comparative study between Xamarin and Cronapp in native and hybrid application development. *Revista Computacao Aplicada*, 2019. vol. 8, no. 1, pp. 13. ISSN 2316-7394.

GRANADOS VALLEJOS, J., *Enfoques y métodos de evaluación de la usabilidad en aplicaciones móviles*. S.I.: Universidad Peruana Unión. 2020.

GUNAWARDHANA, P.D., Native or Web or Hybrid which is better for Mobile Application. *Turkish Journal of Computer and Mathematics Education*, 2021. vol. 12, no. 6, pp. 4643-4649. ISSN 1309-4653.

HERNÁNDEZ SAMPIERI, R., FERNÁNDEZ COLLADO, C. y BAPTISTA LUCIO, P., *Metodología de la investigación*. 6. México D.F.: s.n. 2014. ISBN 9781456223960.

HERNÁNDEZ SAMPIERI, R. y MENDOZA TORRES, C.P., *Metodología de la investigación Las rutas cuantitativa, cualitativa y mixta*. Ciudad de México: s.n. 2018. ISBN 9781456260965.

KIM, T.K. y PARK, J.H., More about the basic assumptions of t-test: Normality and sample size. *Korean Journal of Anesthesiology*, 2019. vol. 72, no. 4, pp. 331-335. ISSN 20057563.

LÉGER, P.M., AN NGUYEN, T., CHARLAND, P., SÉNÉCAL, S., LAPIERRE, H.G. y FREDETTE, M., How Learner Experience and Types of Mobile Applications Influence Performance: The Case of Digital Annotation. *Computers in the Schools*, 2019. vol. 36, no. 2, pp. 83-104. ISSN 1528-7033.

LIM, S.H., Experimental comparison of hybrid and native applications for mobile systems. *International Journal of Multimedia and Ubiquitous Engineering*, 2015. vol. 10, no. 3, pp. 1-12. ISSN 1975-0080.

LOZADA J, 2016, Investigación Aplicada : Definición , Propiedad Intelectual e

- Industria. *Cienciaamérica*, 2014. vol. 1, no. 3, pp. 34-39. ISSN 1390-9592.
- LUPO, T. y BELLOMO, E., A methodological framework based on a danp model for evaluating the software quality in terms of usability: A preliminary investigation on mobile operating systems. *Decision Science Letters*, 2020. vol. 9, no. 4, pp. 521-536. ISSN 1929-5812.
- MA, Y., LIU, X., LIU, Yi, LIU, Yunxin y HUANG, G., A Tale of Two Fashions: An Empirical Study on the Performance of Native Apps and Web Apps on Android. *IEEE Transactions on Mobile Computing*, 2018. vol. 17, no. 5, pp. 990-1003. ISSN 1536-1233.
- MACHINI, A., ENRIQUEZ, J. y CASAS, S., Q2M, a library to compute quality metrics in mobile applications. *Informes Científicos Técnicos - UNPA*, 2019. vol. 11, no. 2, pp. 1-17. ISSN 1852-4516.
- ÑAUPAS, H., PALACIOS, J., ROMERO, H. y VALDIVIA, M. Metodología y diseños en investigación científica. Cuantitativa–Cualitativa y Redacción de la Tesis. 5. Bogotá: Ediciones de la U, 2018. ISBN: 978-958-762-876-0.
- OTZEN, T. y MANTEROLA, C., Sampling Techniques on a Population Study. *International Journal of Morphology*, 2017. vol. 35, no. 1, pp. 227-232. ISSN 0717-9502.
- PANIAGUA L., A., BEDOYA R., D. y MERA, C., A Method for Assessing the Accessibility and Usability in Mobile Applications. *TecnoLógicas*, 2020. vol. 23, no. 48, pp. 99-117. ISSN 0123-7799.
- QUE, P., GUO, X. y ZHU, M., A Comprehensive Comparison between Hybrid and Native App Paradigms. *Proceedings - 2016 8th International Conference on Computational Intelligence and Communication Networks*, 2017. pp. 611-614. ISSN 2472-7555.
- QUIROZ ZAMBRANO, S. y MACÍAS VALENCIA, D., Computer security: considerations. *Dominio de las Ciencias*, 2017. vol. 3, no. 3, pp. 676-688. ISSN 2477-8818.
- ROMERO SALDAÑA, M., Pruebas de bondad de ajuste a una distribución normal. *Enfermería del Trabajo*, 2016. vol. 6, no. 3, pp. 105-114. ISSN

2174-2510.

SAHU, R., Mobile Application Services in Libraries. *Journal of Information Management*, 2018. vol. 3, pp. 44-49. ISSN 2348-1765.

SALGADO ESCOBAR, S., *Estudio De La Eficiencia De Los Frameworks Híbridos Y Nativos En El Desarrollo De Aplicaciones Móviles, Basado En Benchmark Para El Consorcio Informega*. S.l.: Universidad de las fuerzas armadas de Ecuador. 2018.

SÁNCHEZ TURCIOS, R., Prueba de Wilcoxon-Mann-Whitney: mitos y realidades. *Rev Mex Endocrinol Metab Nutr* [en línea], 2015. vol. 2, pp. 18-21. Disponible en: www.endocrinologia.org.mx.

SYAIKHUDDIN, M.M., ANAM, C., RINALDI, A.R. y CONORAS, M.E.B., Conventional Software Testing Using White Box Method. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 2018. vol. 3, no. 1, pp. 65-72. ISSN 2503-2259.

THOMAS, P., DELIA, L., CORBALAN, L., CÁSERES, G., CUITIÑO, A., SOSA, J.F. y PESADO, P., 2017. Análisis de Enfoques de Aplicaciones para Dispositivos Móviles. *Workshop de Investigadores en Ciencias de la Computación*. S.l.: s.n., pp. 428-431. ISBN 978-987-42-5143-5.

THOMAS, P., DELIA, L., CORBALAN, L., CÁSERES, G., SOSA, J.F., TESONE, F., CUITIÑO, A. y PESADO, P., 2018. Tendencias en el desarrollo de Aplicaciones para Dispositivos Móviles. *XX Workshop de Investigadores en Ciencias de la Computación*. S.l.: s.n., pp. 588-592. ISBN 978-987-3619-27-4.

Universidad César Vallejo. 2017. Código de Ética en Investigación. Perú, Trujillo : s.n., 23 de mayo de 2017.

USMAN, M., IQBAL, M.Z. y KHAN, M.U., An automated model-based approach for unit-level performance test generation of mobile applications. *Journal of Software: Evolution and Process*, 2020. vol. 32, no. 1, pp. 1-32. ISSN 2047-7473.

VELÁSQUEZ, S.M., MONSALVE SOSSA, D.E., ZAPATA, M.E., GÓMEZ

- ADASME, M.E. y RÍOS, J.P., Mobile applications testing: advances and challenges. *Lámpsakos*, 2019. vol. 21, no. 21, pp. 39-50. ISSN 2145-4086.
- VENTEU, K.C. y PINTO, G.S., Hybrid mobile development. *Revista Interface Tecnológica*, 2018. vol. 15, no. 1, pp. 11. ISSN 1807-3980.
- VENTURA, J., Population or sample ? A necessary difference. *Revista Cubana de Salud Pública*, 2017. vol. 43, no. 3, pp. 648-649. ISSN 1561-3127.
- WASILEWSKI, K. y WOJCIECH, Z., A Comparison of Java, Flutter and Kotlin/Native Technologies for Sensor Data-Driven Applications. *MDPI*, 2021. pp. 1-16. ISSN 2110-3324. DOI 10.3390.

Anexos

Anexo 1: Matriz de operacionalización de variables

Tabla 26: Matriz de operacionalización de variables

Variable	Definición conceptual	Definición operacional	Dimensión	Indicador	Instrumento	Escala de medición
Rendimiento de aplicaciones móviles	(Thomas et al., 2017, p. 429) señala que debido a las variadas formas de crear aplicaciones móviles es necesario nuevas metodologías y prácticas que beneficien el desarrollo de la ingeniería de software. “El rendimiento es el conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas” (Norma ISO/IEC 9126)	La variable se medirá por medio de una ficha de observación	Tiempo de respuesta (Durán, Peinado y Rosado, 2015)	Tiempo de respuesta (Durán, Peinado y Rosado, 2015)	Ficha de observación (Hernández et al., 2014 p. 252)	Razón (Rodríguez, Pierdant y Rodríguez, 2016)
			Uso de recursos (Durán, Peinado y Rosado, 2015)	Uso de CPU (Durán, Peinado y Rosado, 2015)		
				Uso de la memoria RAM (Durán, Peinado y Rosado, 2015)		
				Uso de la batería (Durán, Peinado y Rosado, 2015)		
			Conexión a la red (Lim, 2015)	Conexión a la red (Lim, 2015)		
Carencia de seguridad (Gunawardhana, 2021)	Carencia de seguridad (Gunawardhana, 2021)					

Anexo 2: Matriz de consistencia

Tabla 27: Matriz de consistencia

Problema general	Objetivo general	Hipótesis general	Variable	Dimensiones	Indicadores
No se ha encontrado una metodología que permita evaluar el rendimiento de las aplicaciones móviles con respecto al uso de recursos, tiempo de respuesta, conexión a la red y carencia de seguridad lo que genera un problema al elegir el tipo de aplicación móvil más adecuado.	Elaborar una metodología para evaluar el rendimiento de las aplicaciones móviles y aplicarla con el fin de determinar que una aplicación móvil nativa tiene mejor rendimiento que una aplicación móvil híbrida.	Haciendo uso de una metodología se determinará que una aplicación nativa tiene mejor rendimiento que una aplicación híbrida con respecto al uso de recursos, tiempo de respuesta, conexión a la red y carencia de seguridad.	rendimiento de aplicaciones móviles	-	-
Problemas específicos	Objetivos específicos	Hipótesis específicas		-	-
No se ha encontrado una metodología que permita evaluar el rendimiento de las aplicaciones móviles con respecto al uso de recursos, lo que genera un problema al elegir el tipo de aplicación móvil más adecuado.	Elaborar una metodología para evaluar el rendimiento de las aplicaciones móviles y aplicarla con el fin de determinar el rendimiento de una aplicación móvil nativa versus la misma aplicación desarrollada con tecnología híbrida teniendo en cuenta el uso de recursos.	Las aplicaciones nativas son más eficientes en cuanto a uso de recursos.		Uso de recursos (Durán, Peinado y Rosado, 2015)	Uso de CPU (Durán, Peinado y Rosado, 2015)
					Uso de la memoria RAM (Durán, Peinado y Rosado, 2015)
					Uso de la batería (Durán, Peinado y

				Rosado, 2015)	
No se ha encontrado una metodología que permita evaluar el rendimiento de las aplicaciones móviles con respecto a los tiempos de respuesta lo que genera un problema al elegir el tipo de aplicación móvil más adecuado.	Elaborar una metodología para evaluar el rendimiento de las aplicaciones móviles y aplicarla con el fin de determinar el rendimiento de una aplicación móvil nativa versus la misma aplicación desarrollada con tecnología híbrida teniendo en cuenta los tiempos de respuesta.	Las aplicaciones nativas son más eficientes en cuanto a tiempos de respuesta.		Tiempo de respuesta (Durán, Peinado y Rosado, 2015)	Tiempo de respuesta (Durán, Peinado y Rosado, 2015)
No se ha encontrado una metodología que permita evaluar el rendimiento de las aplicaciones móviles con respecto a la conexión a la red, lo que genera un problema al elegir el tipo de aplicación móvil más adecuado.	Elaborar una metodología para evaluar el rendimiento de las aplicaciones móviles y aplicarla con el fin de determinar el rendimiento de una aplicación móvil nativa versus la misma aplicación desarrollada con tecnología híbrida teniendo en cuenta la conexión a la red.	Las aplicaciones nativas son más eficientes en cuanto a su conexión a la red.		Conexión a la red (Lim, 2015)	Conexión a la red (Lim, 2015)
No se ha encontrado una metodología que permita evaluar el rendimiento de las aplicaciones móviles con respecto a la carencia de seguridad, lo que genera un problema al elegir el tipo de aplicación móvil más adecuado.	Elaborar una metodología para evaluar el rendimiento de las aplicaciones móviles y aplicarla con el fin de determinar el rendimiento de una aplicación móvil nativa versus la misma aplicación desarrollada con tecnología híbrida teniendo en cuenta la carencia de seguridad.	Las aplicaciones nativas son mejores en cuanto a su carencia de seguridad.		Carencia de seguridad (Gunawardhana, 2021)	Carencia de seguridad (Gunawardhana, 2021)

Anexo 3: Arquitectura utilizada en la investigación

En la arquitectura utilizada en la presente investigación están consideradas dos aplicaciones móviles que comparten las mismas funcionalidades, pero desarrolladas con diferentes tecnologías, una aplicación móvil nativa y una aplicación móvil híbrida. Estas aplicaciones fueron evaluadas con la metodología propuesta “Metodología para la evaluación del rendimiento de aplicaciones móviles” que tiene en cuenta el uso de recursos, tiempo de respuesta, conexión a la red y carencia de seguridad.

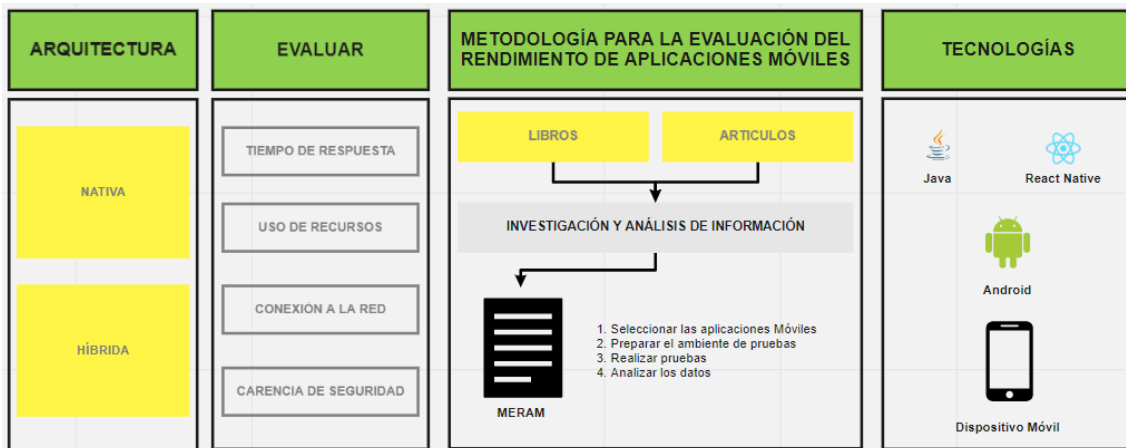


Figura 1. Arquitectura utilizada en la investigación.

Anexo 4: Metodología para la evaluación del rendimiento de aplicaciones móviles

1. Objetivo de la metodología

El objetivo de esta metodología es evaluar el rendimiento de las aplicaciones móviles tanto nativas como híbridas, esta con el propósito de identificar y establecer las diferencias de ellas conforme a los indicadores de uso de recursos, tiempo de respuesta, conexión de red y carencia de seguridad. Para llevar a cabo la metodología, se identificaron las herramientas y/o métodos de evaluación como: Device Info HW, ImmuniWeb, Simple System Monitor y CPU/GPU Meter & Notification.

2. Alcance de la metodología

- Ejecución de procedimiento Pruebas de tiempo de respuesta (Durán, Peinado y Rosado, 2015).
- Ejecución de procedimiento Pruebas de uso de recursos (Durán, Peinado y Rosado, 2015).
- Ejecución de procedimiento Pruebas de conexión de red (Lim, 2015).
- Ejecución de procedimiento Pruebas de carencia de seguridad (Gunawardhana, 2021).

3. Fases de la metodología

La metodología para la evaluación del rendimiento de aplicaciones móviles está dividida en tres fases: Planeación, Implementación y pruebas y Análisis de datos.



Figura 2. Fases de la metodología para la evaluación del rendimiento de aplicaciones móviles.

A continuación, se detalla el desarrollo de las tres fases de la metodología para la evaluación del rendimiento de aplicaciones móviles.

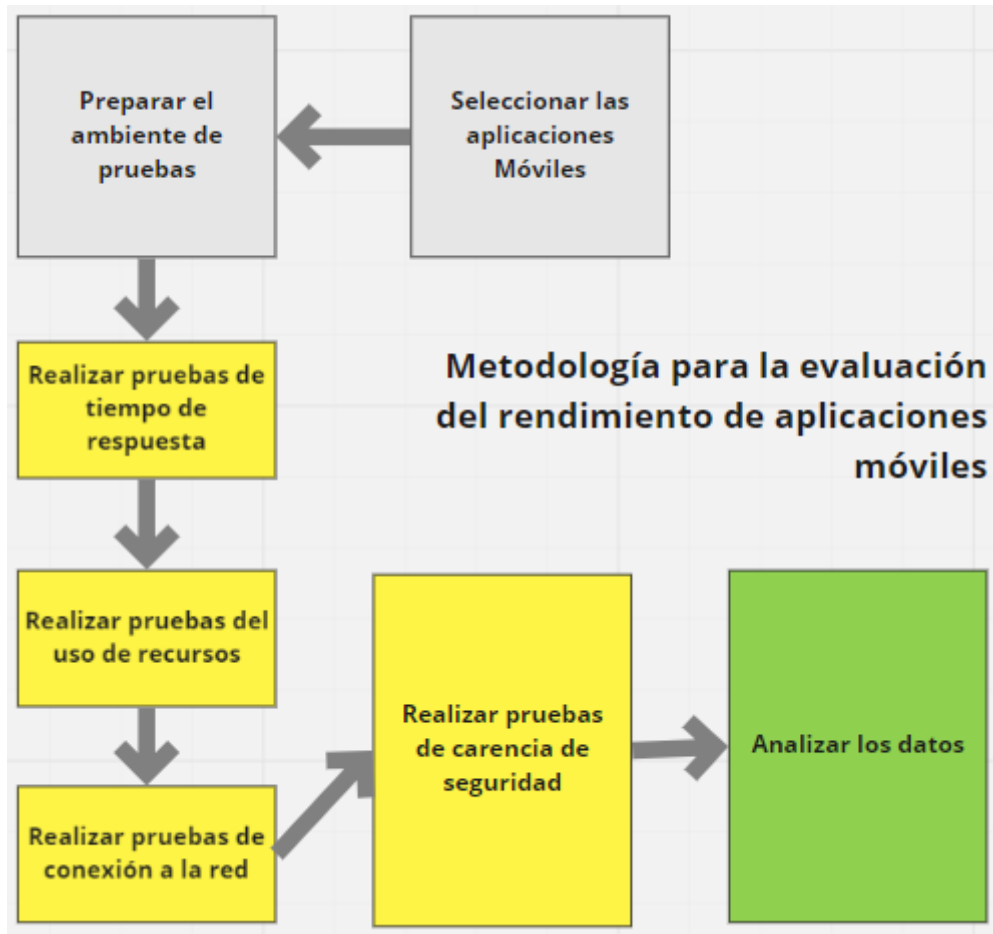


Figura 3. Estructura de las fases de la metodología para la evaluación del rendimiento de aplicaciones móviles.

4. Descripción y procesos

La elaboración de pruebas de evaluación del rendimiento de aplicaciones móviles, no está específico dentro de una metodología, por lo cual la nueva metodología está enfocada en los indicadores que usualmente son tomados en cuenta para una evaluación, Syaikhuddin et al.(2018) indicaron:

El principio de una prueba de aplicación para que se ejecute de forma eficaz y eficiente incluye: 1) Las pruebas deben basarse en los requisitos del usuario; 2) El tiempo y los recursos disponibles para las pruebas son limitados; 3) Los recursos deben utilizarse eficazmente para las pruebas; 4) Las pruebas deben comenzar en pequeños y progresivamente en términos más amplios; 5)

Las pruebas deben ser realizadas por un equipo de examinadores o un equipo externo; 6) Todas las pruebas deben basarse en las necesidades del cliente; 7) Se debe asignar a la mejor persona disponible para realizar las pruebas de software. 8) Los informes de prueba, incluidos los casos de prueba y los informes de prueba para resumir los resultados de las pruebas de la aplicación. 9) El proceso de comprobación del software debe realizarse lo antes posible en el proceso de desarrollo y debe centrarse en la definición del objeto; 10) La planificación de las pruebas debe completarse primero; y 11) El plan de pruebas inicial debe actualizarse a tiempo.

Asimismo, la nueva metodología para la evaluación del rendimiento de aplicaciones móviles empieza especificando los objetivos primordiales, evaluar o comparar dos tecnologías tomando en cuenta el uso de recursos, tiempo de respuesta, conexión a la red y carencia de seguridad.

5. Proceso de desarrollo de la metodología

El proceso o desarrollo para la elaboración de la metodología fueron las siguientes.

- Determinar los objetivos y necesidades del usuario con respecto al rendimiento
- Preparar el ambiente de pruebas para las aplicaciones móviles.
- Ejecutar los procedimientos establecidos en la metodología.

6. Plan de proyecto

Las actividades ejecutadas fueron las siguientes:

- Investigación y recopilación de artículos, tesis y libros virtuales referente a la evaluación y comparación de dos tecnologías móviles.
- Elaboración de una metodología para la evaluación del rendimiento de aplicaciones móviles.
- Aplicación de la metodología para evaluar el rendimiento de aplicaciones móviles.
- Resultado de indicadores.

7. Herramientas y métodos

Para la elaboración de la metodología para la evaluación del rendimiento de aplicaciones móviles se utilizaron ocho herramientas y/o métodos para la evaluación de dos aplicaciones. A continuación, sus descripciones:

A continuación, se muestra la figura 4 donde contiene herramientas y métodos planteados en una nueva arquitectura metodológica. Los recuadros verticales puestos por encima de los pasos, son herramientas y métodos que se emplearán para la evaluación de cada indicador.

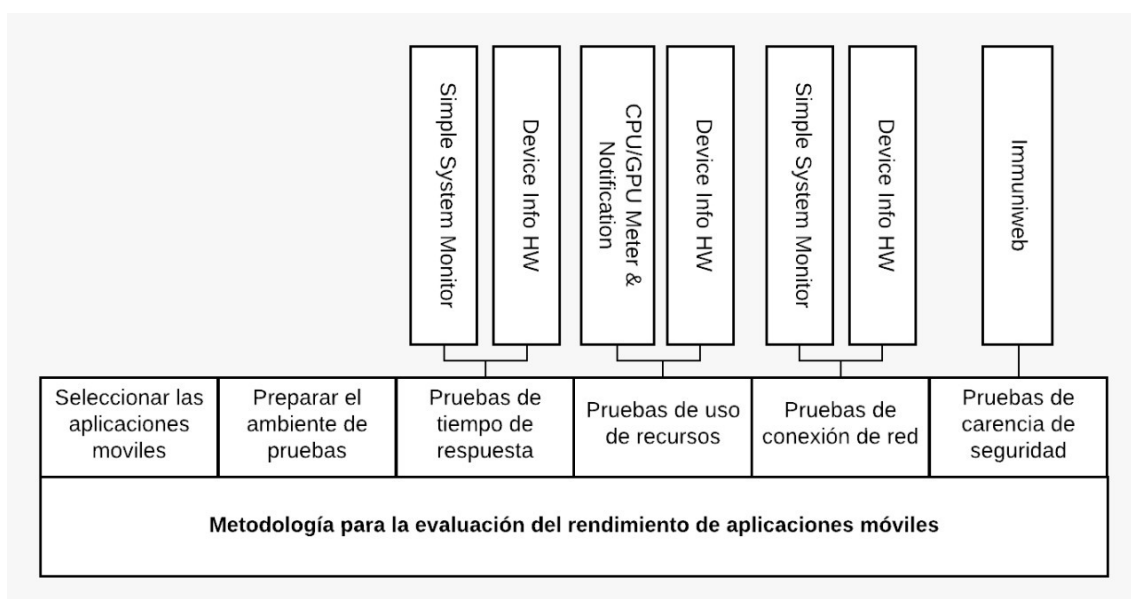


Figura 4. Arquitectura de la metodología.

- **Simple System Monitor:** Es una aplicación móvil desarrollada por Darshan Parajuli, permite monitorizar todos los componentes que tiene el sistema de un dispositivo Android. Dentro de esta se puede visualizar representaciones gráficas de la ejecución del CPU y la frecuencia o constancia de todos los núcleos, tanto el uso de la memoria RAM o GPU.
- **Device Info HW:** Es una aplicación móvil del sistema operativo Android desarrollada por Andrey Efremov, da información de lo lógico (software) y físico (hardware). Esta aplicación identifica los componentes del dispositivo móvil para una lista detallada completa sobre los componentes físicos. Esta es compatible con cámaras, sensores,

batería, flash wi-fi, audio y los demás componentes comúnmente usados para los dispositivos móviles.

- **CPU/GPU Meter & Notification:** Es una aplicación móvil desarrollado por “RAZOR Solutions Ltd”, permite tener todos los datos del CPU o GPU como notificaciones constantes y poder acceder a estos avisos con un solo toque. Esta da en tiempo real el uso de los recursos que consumen la CPU o GPU dando detalles del porcentaje de uso, frecuencia concurrente, frecuencia máxima, temperatura del CPU y la batería y memoria disponible
- **ImmuniWeb:** Es una plataforma web que permite, permite analizar las vulnerabilidades y debilidades de seguridad en aplicaciones móviles. Es un proveedor mundial de administración de attack surface, monitoreo y ejecución de la dark web y servicios de penetración en la seguridad. La plataforma ImmuniWeb mezcla la inteligencia humana con la tecnología de inteligencia artificial.

8. Entradas

Dentro de las entradas se tienen los documentos pertinentes para realizar las pruebas sobre el rendimiento de las aplicaciones móviles. Son las siguientes:

- Establecer objetivos para los procedimientos.
- Determinar herramientas pertinentes para cada una de las pruebas.
- Establecer las aplicaciones a evaluar

9. Proceso de la Metodología

Seguidamente se detalla el proceso de la metodología donde se plasman las entradas, proceso y salidas.

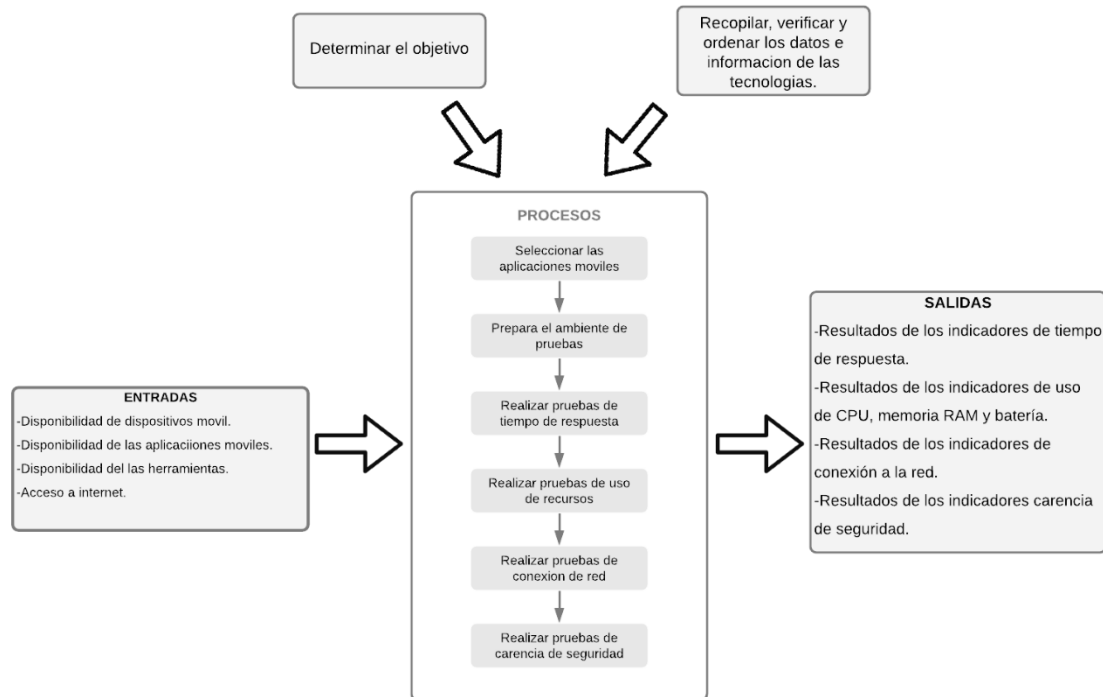


Figura 5. Estructura de los procesos de la metodología para la evaluación del rendimiento de aplicaciones móviles.

10. Salidas

- Los resultados que arroja la metodología son los siguientes:
- Resultados de los indicadores de tiempo de respuesta.
- Resultados de los indicadores de uso de CPU, memoria RAM y batería.
- Resultados de los indicadores de conexión a la red.
- Resultados de los indicadores carencia de seguridad.

11. Procesos específicos de la metodología para evaluar el rendimiento entre una aplicación nativa e híbrida

Proceso 01: Seleccionar las aplicaciones móviles

Objetivo

El objetivo de este proceso es seleccionar las aplicaciones móviles que serán sometidas a pruebas.

Alcance

El alcance de este procedimiento contiene:

- Identificación del tipo de las aplicaciones móviles seleccionadas.
- Identificación de las características de las aplicaciones.
- Descarga o adquisición de las aplicaciones.

Entrada

Para seleccionar las aplicaciones móviles es necesario:

- Disponibilidad de la aplicación, que se encuentre en alguna tienda de aplicaciones móviles o que sea proporcionada por un tercero.

Proceso

En el proceso de selección son necesarias las siguientes actividades:

- Identificar el objetivo de la prueba
- Seleccionar las aplicaciones que serán evaluadas
- Identificar las características de las aplicaciones
- Descargar o adquirir las aplicaciones móviles

Salidas

Los resultados del proceso de selección de las aplicaciones móviles son los siguientes:

Informe de los tipos de aplicaciones

Características de las aplicaciones móviles

Proceso 02: Preparar el ambiente de pruebas

Objetivo

El objetivo de este proceso es preparar el ambiente de pruebas adecuado para realizar la evaluación.

Alcance

El alcance de este procedimiento contiene:

- Identificación de las características del dispositivo móvil (sistema operativo, procesador, memoria RAM, capacidad de la batería, generación de su tecnología móvil y tipo y tamaño de pantalla)

- Especificación de las herramientas necesarias para la ejecución de las pruebas.
- Instalación de las aplicaciones en el dispositivo móvil.

Entrada

Para preparar el ambiente de pruebas es necesario:

- Disponibilidad del dispositivo móvil requerido para la ejecución de las pruebas.
- Disponibilidad de las herramientas y tecnologías necesarias para la ejecución de las pruebas.

Proceso

En el proceso de preparación del ambiente de pruebas son necesarias las siguientes actividades:

- Identificar las características del dispositivo móvil elegido para ejecutar las pruebas teniendo en cuenta (sistema operativo, procesador, memoria RAM, capacidad de la batería, generación de su tecnología móvil y tipo y tamaño de pantalla).
- Instalar las aplicaciones móviles.
- Instalar las herramientas y tecnologías necesarias para la ejecución de las pruebas.

Salidas

Los resultados del proceso de preparación del ambiente de pruebas son los siguientes:

- Aplicaciones instaladas en el dispositivo móvil
- Herramientas instaladas en el dispositivo móvil

Proceso 03: Realizar pruebas de uso de recursos (Durán, Peinado y Rosado, 2015).

Objetivo

El objetivo de este proceso es evaluar el uso de recursos de cada aplicación durante el proceso de prueba.

Alcance

El alcance de este procedimiento contiene:

- Identificación de los objetivos de pruebas
- Descripción de las herramientas utilizadas
- Documentación de la prueba
- Conocimiento del uso de recursos de cada aplicación durante la ejecución de las pruebas.

Entrada

Para realizar las pruebas de tiempo de respuesta es necesario:

- Disponibilidad de las herramientas
- Disponibilidad del dispositivo móvil

Salidas

Los resultados del proceso de prueba de uso de recursos son los siguientes:

- Reporte del indicador uso de procesador.
- Reporte del indicador uso de memoria RAM.
- Reporte de indicador uso de batería

Proceso 04: Realizar pruebas de tiempo de respuesta (Durán, Peinado y Rosado, 2015).

Objetivo

El objetivo de este proceso es evaluar el tiempo de respuesta de cada aplicación móvil.

Alcance

El alcance de este procedimiento contiene:

- Identificación de los objetivos de pruebas
- Descripción de las herramientas utilizadas
- Documentación de la prueba
- Conocimiento de los tiempos de respuesta en cada aplicación móvil

Entrada

Para realizar las pruebas de tiempo de respuesta es necesario:

- Disponibilidad de las herramientas
- Disponibilidad del dispositivo móvil

Proceso

En el proceso de prueba de tiempo de respuesta es necesario realizar las siguientes actividades:

- Planificar el objetivo de la prueba.
- Preparar el ambiente de pruebas.
- Utilizar las herramientas para medir los tiempos de respuesta en cada aplicación.
- Documentar los resultados obtenidos en la prueba

Salidas

Los resultados del proceso de prueba de tiempo de respuesta son los siguientes:

- Reporte del indicador tiempo de respuesta.

Proceso 05: Realizar pruebas de conexión a la red (Lim, 2015).

Objetivo

El objetivo de este proceso es evaluar la conexión a la red de cada aplicación durante el proceso de prueba.

Alcance

El alcance de este procedimiento contiene:

- Identificación de los objetivos de pruebas
- Descripción de las herramientas utilizadas
- Documentación de la prueba
- Conocimiento del estado de la conexión a la red de cada aplicación durante la ejecución de las pruebas.

Entrada

Para realizar las pruebas de conexión a la red es necesario:

- Disponibilidad de las herramientas
- Disponibilidad del dispositivo móvil
- Acceso a Internet

Proceso

En el proceso de prueba de acceso a la red es necesario realizar las siguientes actividades:

- Planificar el objetivo de la prueba.
- Preparar el ambiente de pruebas.
- Utilizar las herramientas para monitorizar la conexión a red en cada aplicación.
- Documentar los resultados obtenidos en la prueba

Salidas

Los resultados del proceso de prueba de acceso a la red son los siguientes:

- Reporte del indicador acceso a la red.

Proceso 06: Realizar pruebas de carencia de seguridad (Gunawardhana, 2021).

Objetivo

El objetivo de este proceso es evaluar la carencia de seguridad de cada aplicación durante el proceso de prueba.

Alcance

El alcance de este procedimiento contiene:

- Identificación de los objetivos de pruebas
- Descripción de las herramientas utilizadas
- Documentación de la prueba
- Conocimiento de la seguridad en cada aplicación durante la ejecución de las pruebas.

Entrada

Para realizar las pruebas de carencia de seguridad es necesario:

- Disponibilidad de las herramientas
- Disponibilidad del dispositivo móvil

Proceso

En el proceso de prueba de carencia de seguridad es necesario realizar las siguientes actividades:

- Planificar el objetivo de la prueba.
- Preparar el ambiente de pruebas.
- Utilizar las herramientas para evaluar la seguridad en cada aplicación.
- Documentar los resultados obtenidos en la prueba

Salidas

Los resultados del proceso de prueba de carencia de seguridad son los siguientes:

- Reporte del indicador carencia de seguridad.

Anexo 5: Descripción de los parámetros de rendimiento

Tabla 28: Descripción de los parámetros de rendimiento por indicadores

Parámetro de rendimiento	Descripción	Indicador
Instalación de la aplicación	Wasilewski y Wojciech (2021) Toma como parámetro de rendimiento el tiempo de compilación y el tamaño del archivo de instalación. Que, Guo y Zhu (2017) Miden el tiempo que toma una aplicación en ser instalada desarrolladas con diferentes paradigmas.	Uso de CPU, uso de memoria RAM, uso de batería, tiempo de respuesta
Inicio de la aplicación	Wasilewski y Wojciech (2021) considera medir el uso de la memoria RAM y CPU durante el inicio de la aplicación para comparar aplicaciones desarrolladas con diferentes tecnologías	Uso de CPU, uso de memoria RAM, uso de batería, tiempo de respuesta
Ejecución de la aplicación	Wasilewski y Wojciech (2021) considera medir el uso de la memoria RAM y CPU durante la ejecución de la aplicación para comparar aplicaciones desarrolladas con diferentes tecnologías	Uso de CPU, uso de memoria RAM, uso de batería, tiempo de respuesta
Ejecución en segundo plano		Uso de CPU, uso de memoria RAM, uso de batería
Carga de archivos	Lim (2015) considera importante evaluar la transferencia de paquetes ya que las aplicaciones comúnmente mas utilizadas son las aplicaciones de redes sociales donde es indispensable la transferencia de paquetes entre usuarios.	Conexión a la red
Descarga de archivos		Conexión a la red
Algoritmos de Hash débiles	Los algoritmos de hash débiles (por ejemplo, MD2, MD4, MD5 o SHA-1) pueden ser vulnerables a colisiones y otras debilidades de seguridad, y no deben usarse cuando se requiere un hash de datos confiable.	Carencia de seguridad
Datos codificados	La aplicación móvil contiene información técnica o de depuración que puede ser extraída y utilizada por un atacante para facilitar más ataques.	Carencia de seguridad
Modo depuración habilitado	Este modo puede exponer información técnica y puede facilitar la ingeniería inversa de la aplicación.	Carencia de seguridad
Copia de seguridad habilitada	La aplicación móvil utiliza una función de copia de seguridad externa que puede almacenar datos confidenciales de la aplicación. En determinadas condiciones, esto puede dar lugar a la divulgación de información (por ejemplo, cuando un servidor de respaldo o su cuenta de Gmail están comprometidos).	Carencia de seguridad
JS habilitador en webview	La aplicación móvil ha habilitado JavaScript en WebView. De forma predeterminada, JavaScript está deshabilitado en WebView; si está	Carencia de seguridad

	habilitado, puede traer varios problemas de seguridad relacionados con JS, como ataques de Cross-Site Scripting (XSS).	
Falta protección contra tomas	El sistema operativo Android permite que una aplicación móvil muestre su interfaz de usuario sobre la interfaz de usuario de otra aplicación instalada y ejecutada en el dispositivo. Cuando el usuario toca la pantalla, la aplicación puede pasar el evento táctil a otra aplicación debajo de su capa de interfaz de usuario que el usuario no ve, sirviendo como un proxy para pasar actividades táctiles no deseadas. Este ataque es bastante similar al clickjacking pero para dispositivos móviles. Para ser explotado con éxito, una aplicación maliciosa ya debe estar instalada en el teléfono móvil de la víctima. Un ejemplo de explotación sería una aplicación de malware que engaña al usuario para que toque involuntariamente un botón de pago (o cualquier otra funcionalidad) de una aplicación sensible cuando juega un juego o realiza otra actividad inocente en la pantalla de la aplicación maliciosa.	Carencia de seguridad
Actividades exportadas	Pueden ser cargadas y ejecutadas por otras aplicaciones que residen en el dispositivo móvil, incluidas las maliciosas, para activar una actividad de aplicación legítima. Una actividad es un componente de Android que permite interactuar con la aplicación de una forma determinada (por ejemplo, realizar determinadas acciones o funciones).	Carencia de seguridad
Creación de archivo temporal	La aplicación móvil crea archivos temporales. A pesar de que los archivos de caché suelen ser privados de forma predeterminada, se recomienda asegurarse de que los archivos temporales se eliminen de forma segura cuando la aplicación ya no los necesite.	Carencia de seguridad
Uso de intención implícita	Las intenciones permiten que las aplicaciones móviles se comuniquen entre sí solicitando realizar diferentes acciones para las que son más adecuadas. Sin embargo, una intención implícita no especifica a qué aplicación en particular envía una solicitud para realizar una acción. Si se instala una aplicación maliciosa en el dispositivo de la víctima, también puede recibir la intención implícita, responder a ella y realizar alguna acción en su lugar, o además de, una aplicación legítima.	Carencia de seguridad
Uso del filtro de intención	Puede suponer un grave riesgo de seguridad si no se implementa y filtra correctamente. Los desarrolladores no deben confiar únicamente en los filtros de intenciones por motivos de seguridad porque no imponen restricciones a las intenciones explícitas. Los filtros de intenciones se definen en el archivo de manifiesto de Android y permiten a los desarrolladores elegir qué tipo de intenciones deben recibir y gestionar sus componentes de aplicación.	Carencia de seguridad

Carga dinámica de código	En determinadas circunstancias, la carga dinámica de código puede ser peligrosa. Por ejemplo, si el código está ubicado en un almacenamiento externo (por ejemplo, una tarjeta SD), esto puede conducir a una vulnerabilidad de inyección de código si el almacenamiento externo es legible y / o escribible y un atacante puede acceder a él.	Carencia de seguridad
Deserialización de objeto encontrada	La deserialización de objetos realizada en un recurso que no es de confianza (por ejemplo, entrada proporcionada por el usuario o almacenamiento externo) puede ser peligrosa si un atacante manipula los datos para la deserialización.	Carencia de seguridad
Falta anti-emulación	La aplicación móvil no utiliza ninguna técnica anti-emulación o anti-depurador (por ejemplo, detectar dispositivos roteados o comprobar si los contactos son auténticos). Esto puede facilitar significativamente los procesos de depuración e ingeniería inversa de aplicaciones.	Carencia de seguridad
La configuración de seguridad de red no está presente	La aplicación móvil no usa la configuración de seguridad de la red para definir qué certificados y autoridades de certificación (CA) se pueden usar para diferentes entornos (por ejemplo, desarrollo, prueba y producción). La función Configuración de seguridad de red en Android permite a los desarrolladores de aplicaciones personalizar la configuración de seguridad de su red en un archivo de configuración declarativo y seguro sin modificar el código de la aplicación.	Carencia de seguridad

Anexo 6: Resultados de la prueba de uso del CPU

Tabla 29: Ficha de recopilación de datos – FORM-01 del indicador uso del CPU

FORM-01		Proceso 03: Realizar pruebas de uso de recursos	
Indicador uso de CPU			
Herramientas: CPU/GPU Meter & Notification, Device info HW			
Parámetro de rendimiento	Aplicación Móvil Nativa - (% de uso de CPU)	Aplicación Móvil Híbrida - (% de uso de CPU)	
Instalación de la aplicación	28	34	
Instalación de la aplicación	24	40	
Instalación de la aplicación	21	37	
Instalación de la aplicación	30	34	
Instalación de la aplicación	20	34	
Instalación de la aplicación	21	29	

Instalación de la aplicación	38	40
Instalación de la aplicación	37	44
Instalación de la aplicación	30	39
Instalación de la aplicación	22	28
Instalación de la aplicación	21	32
Instalación de la aplicación	29	31
Instalación de la aplicación	35	34
Instalación de la aplicación	28	30
Instalación de la aplicación	27	33
Instalación de la aplicación	35	34
Instalación de la aplicación	31	37
Instalación de la aplicación	32	38
Instalación de la aplicación	29	31
Instalación de la aplicación	27	30
Iniciar la aplicación	24	30
Iniciar la aplicación	30	28
Iniciar la aplicación	25	25
Iniciar la aplicación	23	30
Iniciar la aplicación	28	27
Iniciar la aplicación	24	24
Iniciar la aplicación	30	29
Iniciar la aplicación	22	31
Iniciar la aplicación	37	36
Iniciar la aplicación	25	34
Iniciar la aplicación	32	30
Iniciar la aplicación	36	31
Iniciar la aplicación	26	39
Iniciar la aplicación	24	36
Iniciar la aplicación	31	38
Iniciar la aplicación	27	31
Iniciar la aplicación	28	30
Iniciar la aplicación	26	28
Iniciar la aplicación	32	36
Iniciar la aplicación	25	42
Ejecutar aplicación	35	31
Ejecutar aplicación	38	37
Ejecutar aplicación	26	36
Ejecutar aplicación	24	34
Ejecutar aplicación	27	29

Anexo 7: Resultados de la prueba de uso de memoria RAM

Tabla 30: Ficha de recopilación de datos – FORM-02 del indicador uso de memoria RAM

FORM-02	Proceso 03: Realizar pruebas de uso de recursos	
Indicador uso de memoria RAM		
Herramientas: CPU/GPU Meter & Notification, Device info HW		
Parámetro de rendimiento	Aplicación Móvil Nativa (Uso de memoria RAM - MB)	Aplicación Móvil Híbrida (Uso de memoria RAM - MB)
Instalación de la aplicación	15	16
Instalación de la aplicación	15	15
Instalación de la aplicación	17	17
Instalación de la aplicación	16	17
Instalación de la aplicación	16	18
Instalación de la aplicación	16	15
Instalación de la aplicación	15	18
Instalación de la aplicación	16	16
Instalación de la aplicación	15	16
Instalación de la aplicación	17	17
Instalación de la aplicación	18	21
Instalación de la aplicación	14	19
Instalación de la aplicación	15	16
Instalación de la aplicación	15	16
Instalación de la aplicación	15	17
Instalación de la aplicación	16	18
Instalación de la aplicación	16	18
Instalación de la aplicación	17	15
Instalación de la aplicación	15	16
Instalación de la aplicación	16	16
Iniciar la aplicación	18	22
Iniciar la aplicación	15	15
Iniciar la aplicación	14	16
Iniciar la aplicación	14	14
Iniciar la aplicación	15	15
Iniciar la aplicación	16	16
Iniciar la aplicación	15	15
Iniciar la aplicación	15	16
Iniciar la aplicación	14	16

Iniciar la aplicación	14	16
Iniciar la aplicación	15	18
Iniciar la aplicación	14	17
Iniciar la aplicación	14	17
Iniciar la aplicación	16	16
Iniciar la aplicación	14	16
Iniciar la aplicación	14	18
Iniciar la aplicación	15	16
Iniciar la aplicación	14	15
Iniciar la aplicación	15	17
Iniciar la aplicación	12	16
Ejecutar aplicación	18	22
Ejecutar aplicación	18	20
Ejecutar aplicación	17	19
Ejecutar aplicación	15	17
Ejecutar aplicación	16	16
Ejecutar aplicación	20	19
Ejecutar aplicación	22	21
Ejecutar aplicación	24	25
Ejecutar aplicación	26	27
Ejecutar aplicación	24	23
Ejecutar aplicación	23	24
Ejecutar aplicación	25	23
Ejecutar aplicación	20	25
Ejecutar aplicación	20	25
Ejecutar aplicación	22	22
Ejecutar aplicación	24	23
Ejecutar aplicación	21	25
Ejecutar aplicación	22	24
Ejecutar aplicación	21	24
Ejecutar aplicación	23	26
Aplicación en segundo plano	5	5
Aplicación en segundo plano	5	6
Aplicación en segundo plano	4	5
Aplicación en segundo plano	6	4
Aplicación en segundo plano	5	5
Aplicación en segundo plano	4	4
Aplicación en segundo plano	4	5
Aplicación en segundo plano	5	4

Aplicación en segundo plano	5	5
Aplicación en segundo plano	4	4
Aplicación en segundo plano	3	4
Aplicación en segundo plano	5	6
Aplicación en segundo plano	4	4
Aplicación en segundo plano	4	5
Aplicación en segundo plano	4	4
Aplicación en segundo plano	4	6
Aplicación en segundo plano	4	5
Aplicación en segundo plano	5	4
Aplicación en segundo plano	4	4
Aplicación en segundo plano	4	4

Anexo 8: Resultados de la prueba de uso de la batería

Tabla 31: Ficha de recopilación de datos – FORM-04 del indicador uso de la batería

FORM-03	Proceso 03: Realizar pruebas de uso de recursos	
Indicador uso de la batería		
Herramientas: CPU/GPU Meter & Notification, Device info HW		
Parámetro de rendimiento	Aplicación Móvil Nativa - (% de uso de Bateria)	Aplicación Móvil Híbrida - (% de uso de Bateria)
Uso (10 minutos)	1	1
Uso (20 minutos)	2	3
Uso (30 minutos)	6	9
Uso (60 minutos)	14	16
Segundo plano (1 hora)	0	0
Segundo plano (2 horas)	0	0
Segundo plano (3 horas)	1	1
Segundo plano (4 horas)	1	1
Segundo plano (5 horas)	1	1
Segundo plano (6 horas)	1	2
Segundo plano (7 horas)	2	2
Segundo plano (8 horas)	2	2
Segundo plano (9 horas)	2	3
Segundo plano (10 horas)	2	3
Segundo plano (11 horas)	3	3

Segundo plano (12 horas)	3	4
Segundo plano (24 horas)	5	7

Anexo 9: Resultados de la prueba de tiempo de respuesta

Tabla 32: Ficha de recopilación de datos – FORM-04 del indicador tiempo de respuesta

FORM-04		Proceso 04: Realizar pruebas de tiempo de respuesta	
Indicador tiempo de respuesta			
Herramientas: Simple System Monitor, Device info HW			
Parámetro de rendimiento	Aplicación Móvil Nativa - Tiempo (ms)	Aplicación Móvil Híbrida - Tiempo (ms)	
Instalación de la aplicación	8179	9277	
Instalación de la aplicación	8401	9292	
Instalación de la aplicación	8726	8962	
Instalación de la aplicación	8417	9271	
Instalación de la aplicación	8747	9034	
Instalación de la aplicación	8922	9113	
Instalación de la aplicación	8740	9168	
Instalación de la aplicación	8643	9157	
Instalación de la aplicación	8155	9380	
Instalación de la aplicación	8067	9030	
Instalación de la aplicación	8513	9549	
Instalación de la aplicación	8435	9276	
Instalación de la aplicación	8400	9033	
Instalación de la aplicación	8330	9031	
Instalación de la aplicación	8569	9192	
Instalación de la aplicación	8440	9332	
Instalación de la aplicación	8522	9417	
Instalación de la aplicación	8436	9336	
Instalación de la aplicación	8379	9684	
Instalación de la aplicación	8350	9521	
Iniciar la aplicación	2140	2304	
Iniciar la aplicación	1610	1637	
Iniciar la aplicación	1560	1530	
Iniciar la aplicación	1523	1647	

Iniciar la aplicación	1498	1654
Iniciar la aplicación	1536	1647
Iniciar la aplicación	1485	1671
Iniciar la aplicación	1537	1540
Iniciar la aplicación	1421	1502
Iniciar la aplicación	1498	1621
Iniciar la aplicación	1599	1612
Iniciar la aplicación	1515	1592
Iniciar la aplicación	1594	1503
Iniciar la aplicación	1586	1608
Iniciar la aplicación	1439	1667
Iniciar la aplicación	1440	1501
Iniciar la aplicación	1590	1506
Iniciar la aplicación	1504	1648
Iniciar la aplicación	1588	1600
Iniciar la aplicación	1536	1657
Ejecutar aplicación	4640	5697
Ejecutar aplicación	4590	4833
Ejecutar aplicación	4676	5076
Ejecutar aplicación	4507	5387
Ejecutar aplicación	4266	4787
Ejecutar aplicación	4084	4904
Ejecutar aplicación	4383	4399
Ejecutar aplicación	4624	4510
Ejecutar aplicación	4398	4501
Ejecutar aplicación	4592	4133
Ejecutar aplicación	4341	4532
Ejecutar aplicación	4074	5045
Ejecutar aplicación	4502	4577
Ejecutar aplicación	4047	4349
Ejecutar aplicación	4447	5038
Ejecutar aplicación	4427	4817
Ejecutar aplicación	4080	4704
Ejecutar aplicación	4388	4863
Ejecutar aplicación	4373	4448
Ejecutar aplicación	4530	4610

Anexo 10: Resultados de la prueba de conexión a la red

Tabla 33: Ficha de recopilación de datos – FORM-05 del indicador conexión a la red

FORM-05	Proceso 05: Realizar pruebas de conexión a la red	
Indicador acceso a la red		
Herramientas: Simple System Monitor y Device Info HW		
Parámetro de rendimiento	Aplicación Móvil Nativa - Tiempo (ms)	Aplicación Móvil Híbrida - Tiempo (ms)
Cargar - 512 KB	2967	3408
Cargar - 512 KB	3041	2784
Cargar - 512 KB	2829	2720
Cargar - 512 KB	2694	2536
Cargar - 512 KB	2480	2567
Cargar - 512 KB	2763	2885
Cargar - 512 KB	2524	2709
Cargar - 512 KB	2438	2765
Cargar - 512 KB	2794	2856
Cargar - 512 KB	2603	2837
Cargar - 512 KB	2731	2774
Cargar - 512 KB	2531	2807
Cargar - 512 KB	2644	2779
Cargar - 512 KB	2631	2692
Cargar - 512 KB	2533	2820
Cargar - 512 KB	2656	2603
Cargar - 512 KB	2745	2594
Cargar - 512 KB	2626	2748
Cargar - 512 KB	2575	2760
Cargar - 512 KB	2476	2612
Descargar - 512 KB	928	1212
Descargar - 512 KB	2118	785
Descargar - 512 KB	867	1996
Descargar - 512 KB	918	924
Descargar - 512 KB	1002	1102
Descargar - 512 KB	866	910
Descargar - 512 KB	886	917
Descargar - 512 KB	970	970
Descargar - 512 KB	887	1008

Descargar - 512 KB	911	965
Descargar - 512 KB	982	946
Descargar - 512 KB	956	924
Descargar - 512 KB	983	996
Descargar - 512 KB	985	1017
Descargar - 512 KB	910	1040
Descargar - 512 KB	950	953
Descargar - 512 KB	928	926
Descargar - 512 KB	974	1071
Descargar - 512 KB	948	1027
Descargar - 512 KB	957	1011
Cargar - 1 MB	2763	2984
Cargar - 1 MB	2607	2648
Cargar - 1 MB	2648	2687
Cargar - 1 MB	2993	2528
Cargar - 1 MB	2514	2631
Cargar - 1 MB	2738	2602
Cargar - 1 MB	2764	2745
Cargar - 1 MB	2740	2601
Cargar - 1 MB	2686	2756
Cargar - 1 MB	2771	2836
Cargar - 1 MB	2840	2707
Cargar - 1 MB	2860	2737
Cargar - 1 MB	2808	2772
Cargar - 1 MB	2861	2800
Cargar - 1 MB	2756	2763
Cargar - 1 MB	2748	2918
Cargar - 1 MB	2772	2732
Cargar - 1 MB	2743	2921
Cargar - 1 MB	2627	2775
Cargar - 1 MB	2790	2918
Descargar - 1 MB	1604	2193
Descargar - 1 MB	1169	1088
Descargar - 1 MB	1134	1368
Descargar - 1 MB	2002	1424
Descargar - 1 MB	1211	1218
Descargar - 1 MB	1177	1182
Descargar - 1 MB	1209	1220
Descargar - 1 MB	1153	1255

Descargar - 1 MB	1178	1261
Descargar - 1 MB	1202	1211
Descargar - 1 MB	1152	1213
Descargar - 1 MB	1193	1172
Descargar - 1 MB	1159	1248
Descargar - 1 MB	1182	1170
Descargar - 1 MB	1175	1271
Descargar - 1 MB	1174	1159
Descargar - 1 MB	1154	1178
Descargar - 1 MB	1177	1202
Descargar - 1 MB	1169	1247
Descargar - 1 MB	1149	1258
Cargar - 5 MB	3911	4952
Cargar - 5 MB	4195	4500
Cargar - 5 MB	3766	3674
Cargar - 5 MB	3793	3812
Cargar - 5 MB	3724	3721
Cargar - 5 MB	3722	3800
Cargar - 5 MB	3724	3785
Cargar - 5 MB	3770	3790
Cargar - 5 MB	3773	3762
Cargar - 5 MB	3771	3773
Cargar - 5 MB	3761	3856
Cargar - 5 MB	3676	3874
Cargar - 5 MB	3755	3772
Cargar - 5 MB	3661	3712
Cargar - 5 MB	3770	3735
Cargar - 5 MB	3734	3765
Cargar - 5 MB	3657	3727
Cargar - 5 MB	3700	3865
Cargar - 5 MB	3673	3842
Cargar - 5 MB	3659	3717
Descargar - 5 MB	1616	3377
Descargar - 5 MB	2088	2128
Descargar - 5 MB	1729	1813
Descargar - 5 MB	1722	2001
Descargar - 5 MB	1750	1894
Descargar - 5 MB	1729	1751
Descargar - 5 MB	1758	1764

Descargar - 5 MB	1721	1833
Descargar - 5 MB	1727	1792
Descargar - 5 MB	1751	1734
Descargar - 5 MB	1775	1742
Descargar - 5 MB	1768	1885
Descargar - 5 MB	1757	1843
Descargar - 5 MB	1740	1782
Descargar - 5 MB	1762	1804
Descargar - 5 MB	1787	1825
Descargar - 5 MB	1778	1852
Descargar - 5 MB	1769	1756
Descargar - 5 MB	1749	1822
Descargar - 5 MB	1759	1800
Cargar - 10 MB	5168	6168
Cargar - 10 MB	4330	5400
Cargar - 10 MB	4328	4512
Cargar - 10 MB	4634	5012
Cargar - 10 MB	4521	5302
Cargar - 10 MB	4480	4565
Cargar - 10 MB	4528	4382
Cargar - 10 MB	4470	4481
Cargar - 10 MB	4350	4575
Cargar - 10 MB	4404	4440
Cargar - 10 MB	4354	4657
Cargar - 10 MB	4370	4482
Cargar - 10 MB	4384	5056
Cargar - 10 MB	4525	5082
Cargar - 10 MB	4471	4659
Cargar - 10 MB	4437	4899
Cargar - 10 MB	4487	4522
Cargar - 10 MB	4441	5113
Cargar - 10 MB	4428	4525
Cargar - 10 MB	4448	4650
Descargar - 10 MB	2632	2987
Descargar - 10 MB	3125	4331
Descargar - 10 MB	3213	2990
Descargar - 10 MB	2874	3153
Descargar - 10 MB	2678	2987
Descargar - 10 MB	2562	2612

Descargar - 10 MB	3102	2916
Descargar - 10 MB	3012	3242
Descargar - 10 MB	3198	3254
Descargar - 10 MB	2812	3164
Descargar - 10 MB	2803	3062
Descargar - 10 MB	2842	2804
Descargar - 10 MB	2611	3212
Descargar - 10 MB	2585	2977
Descargar - 10 MB	2569	2923
Descargar - 10 MB	3006	3133
Descargar - 10 MB	2672	3062
Descargar - 10 MB	3175	3166
Descargar - 10 MB	3061	2682
Descargar - 10 MB	2819	2710
Cargar - 20 MB	8168	8160
Cargar - 20 MB	7700	7679
Cargar - 20 MB	7382	7983
Cargar - 20 MB	7494	7859
Cargar - 20 MB	7619	7998
Cargar - 20 MB	7555	7919
Cargar - 20 MB	7786	7652
Cargar - 20 MB	7965	7651
Cargar - 20 MB	7764	7764
Cargar - 20 MB	7332	7949
Cargar - 20 MB	7788	7990
Cargar - 20 MB	7675	7870
Cargar - 20 MB	7934	7648
Cargar - 20 MB	7951	7799
Cargar - 20 MB	7769	7988
Cargar - 20 MB	7563	7844
Cargar - 20 MB	7904	7828
Cargar - 20 MB	7459	7854
Cargar - 20 MB	7549	7731
Cargar - 20 MB	7738	7746
Descargar - 20 MB	5471	5308
Descargar - 20 MB	5089	5334
Descargar - 20 MB	5153	4915
Descargar - 20 MB	4770	5223
Descargar - 20 MB	5484	5374

Descargar - 20 MB	4699	5751
Descargar - 20 MB	5059	5536
Descargar - 20 MB	4613	5834
Descargar - 20 MB	5062	5832
Descargar - 20 MB	5293	5832
Descargar - 20 MB	4711	5305
Descargar - 20 MB	5408	5837
Descargar - 20 MB	5106	4867
Descargar - 20 MB	5285	5368
Descargar - 20 MB	5343	5191
Descargar - 20 MB	4765	5895
Descargar - 20 MB	4629	5045
Descargar - 20 MB	5325	5159
Descargar - 20 MB	4693	5199
Descargar - 20 MB	4999	4953

Anexo 11: Resultados de la prueba de carencia de seguridad

Tabla 34: Ficha de recopilación de datos – FORM-06 del indicador carencia de seguridad

FORM06	Proceso 06: Realizar pruebas de carencia de seguridad	
Indicador carencia de seguridad		
Herramientas: Immuniweb		
Parámetro de rendimiento	Aplicación Móvil Nativa (Nro. De vulnerabilidades encontradas)	Aplicación Móvil Híbrida (Nro. De vulnerabilidades encontradas)
Algoritmos de Hash débiles	0	5
Datos codificados	0	2
Modo depuración habilitado	3	0
Copia de seguridad habilitada	3	0
JS habilitador en webview	3	0
Falta protección contra tomas	2	2
Actividades exportadas	2	2
Creación de archivo temporal	1	1
Uso de intención implícita	1	0
Uso del filtro de intención	1	1
Carga dinámica de código	1	1
Deserialización de objeto encontrada	1	1
Falta anti-emulacion	1	1
La configuración de seguridad de red no está presente	1	1

Anexo 12: Características y funcionalidades de las aplicaciones sometidas a pruebas

Las aplicaciones móviles se desarrollaron utilizando diferentes tecnologías, una con tecnología nativa haciendo uso del lenguaje de programación java y la otra aplicación será desarrollada con tecnología híbrida haciendo uso de JavaScript y React Native. Ambas aplicaciones tienen las mismas funcionalidades y utilizan Firebase para las sesiones, transferencia de datos y notificaciones.

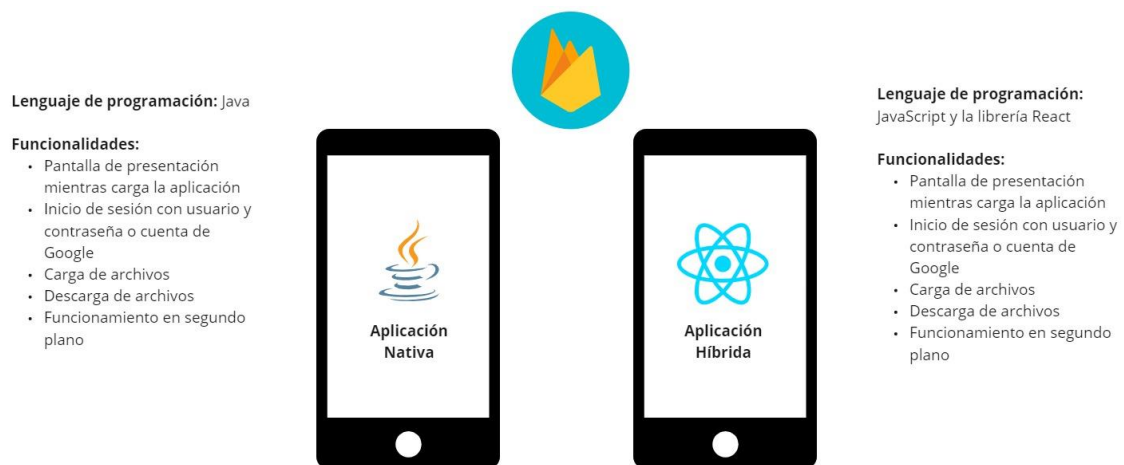


Figura 6. Características y funcionalidades necesarias de las aplicaciones que fueron sometidas a pruebas.

La instalación de las aplicaciones se realizó en el mismo dispositivo para garantizar la igualdad de condiciones. Durán, Peinado y Rosado (2015) indicaron que antes de iniciar la evaluación es necesario obtener la información del uso actual de la batería, memoria RAM y CPU. (p. 85). Esto permitirá conocer las condiciones iniciales del dispositivo móvil previas a la evaluación.

Anexo 13: Aplicación de la metodología MERAM

Proceso 01: Seleccionar las aplicaciones móviles

Las aplicaciones seleccionadas fueron una aplicación móvil nativa y una aplicación móvil híbrida ver (Anexo 14).

Proceso 02: Preparar el ambiente de pruebas

Instalación de las herramientas y tecnologías necesarias para la ejecución de las pruebas.

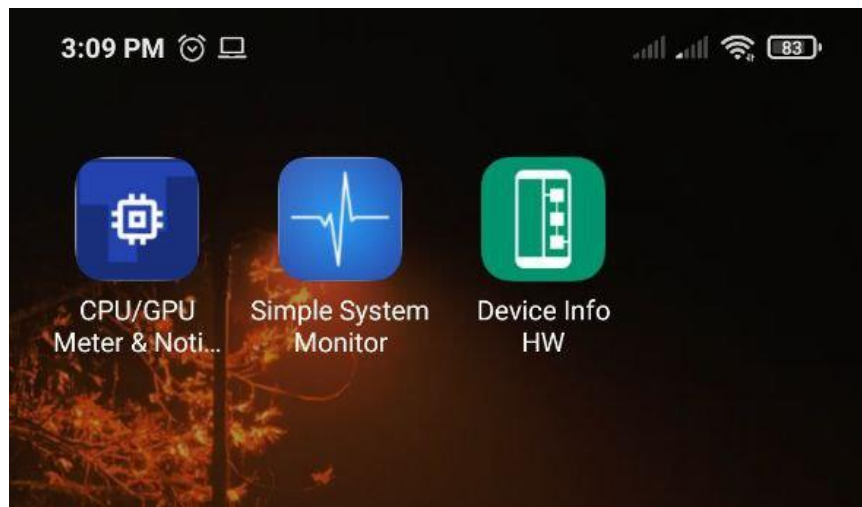


Figura 7. Herramientas utilizadas para las pruebas de uso de recursos, tiempo de respuesta, conexión a la red y carencia de seguridad (Device Info HW, Simple System Monitor y CPU/GPU Meter & Notification).

Identificación de las características del dispositivo móvil elegido para ejecutar las pruebas.

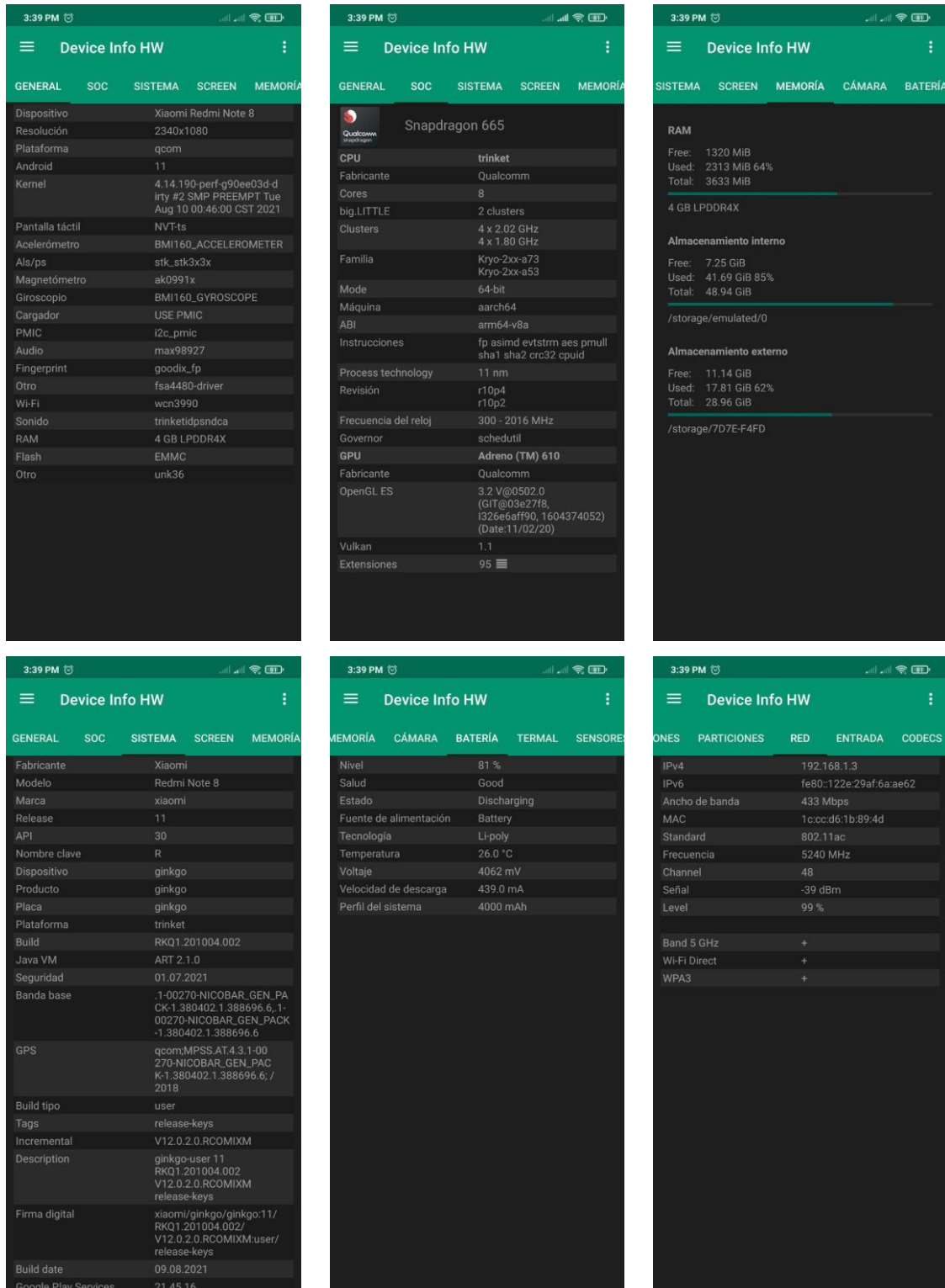


Figura 8. Identificación de las características del dispositivo utilizando la herramienta Device Info HW).

Instalación de las aplicaciones móviles que fueron sometidas a pruebas de rendimiento.

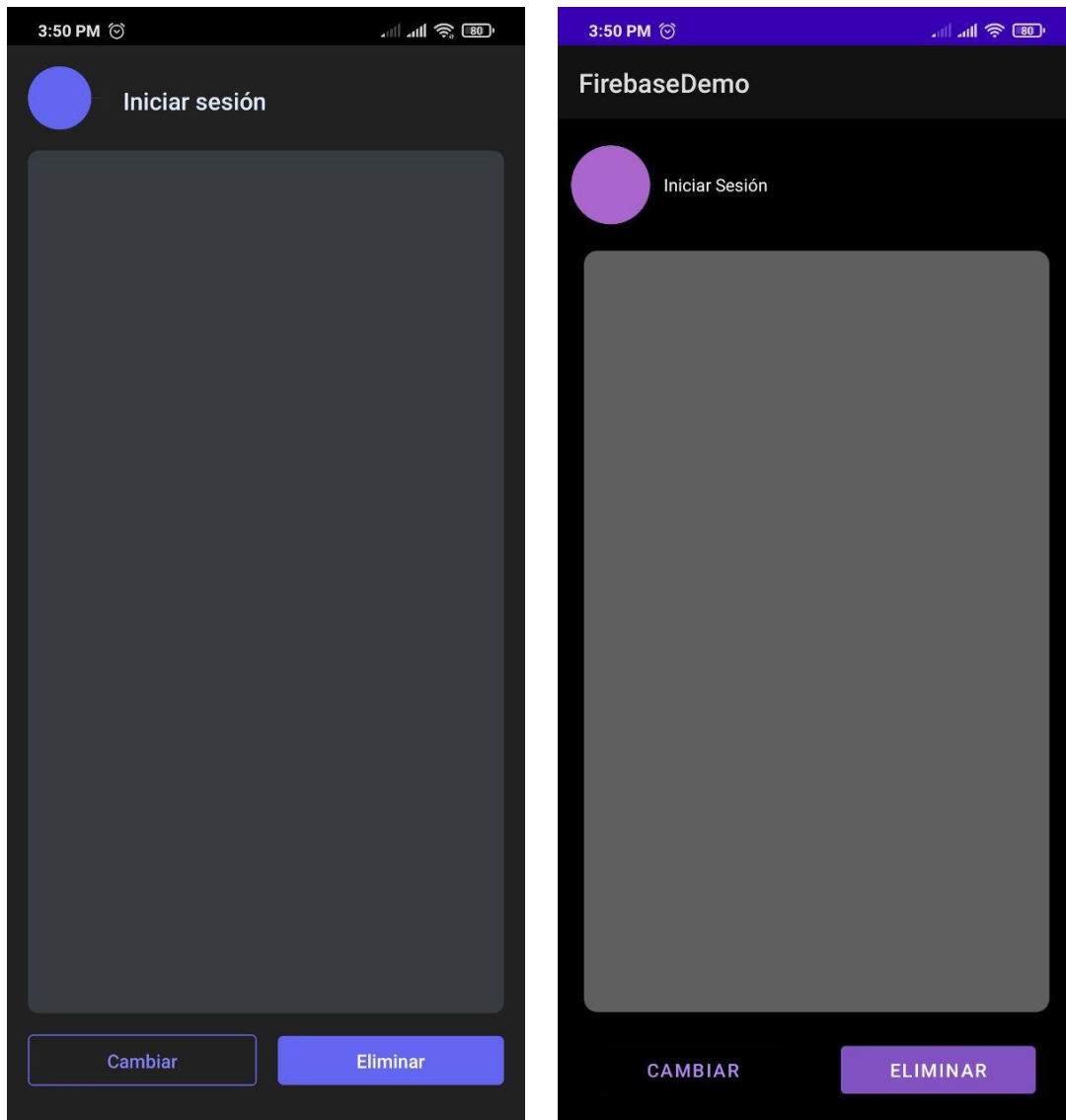


Figura 9. Instalación de las aplicaciones móviles.

Proceso 03: Realizar pruebas de uso de recursos

Ejecución de pruebas de rendimiento de uso de CPU con la herramienta CPU/GPU Meter & Notification durante la instalación, inicio y ejecución de las aplicaciones, una vez generado el reporte los datos fueron almacenados para su posterior análisis.

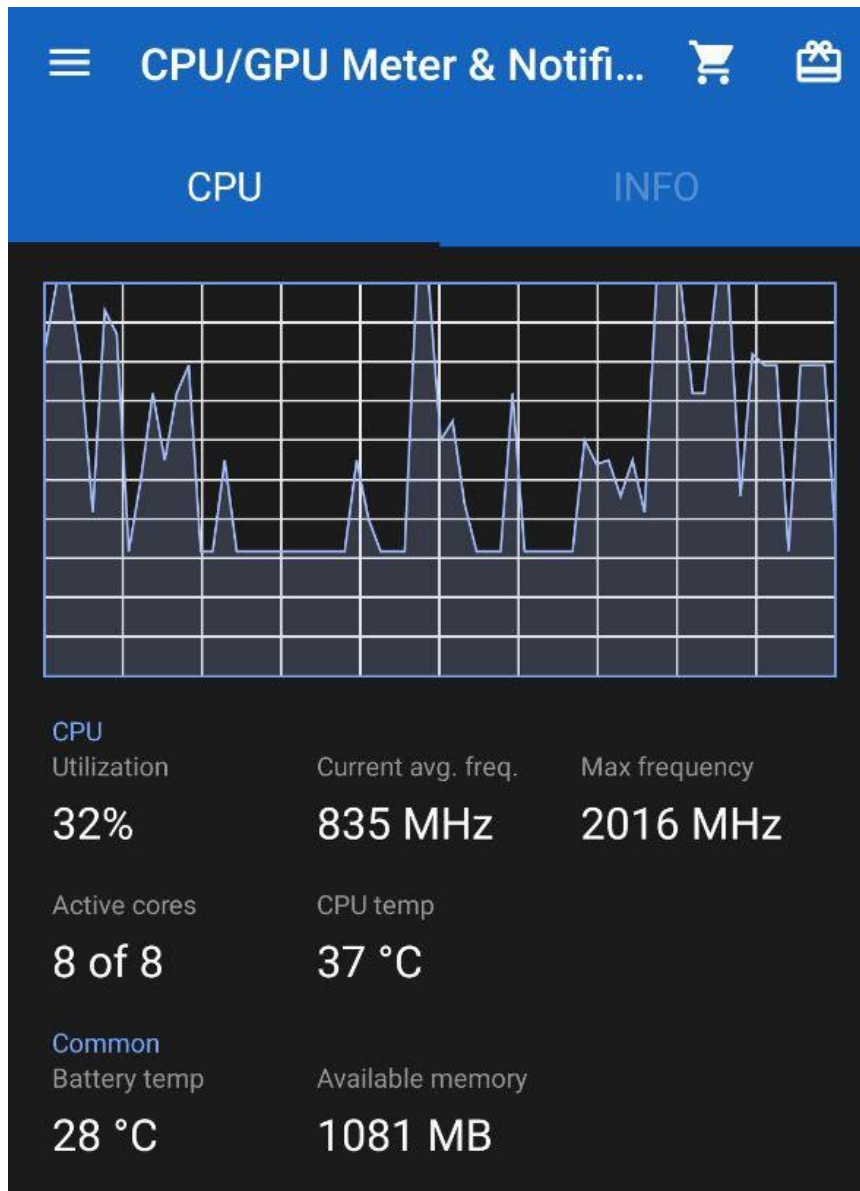


Figura 10. Ejecución de pruebas de rendimiento uso de CPU.

Ejecución de pruebas de rendimiento de uso de memoria RAM con la herramienta Simple System Monitor durante la instalación, inicio y ejecución de las aplicaciones, una vez generado el reporte los datos fueron almacenados para su posterior análisis.

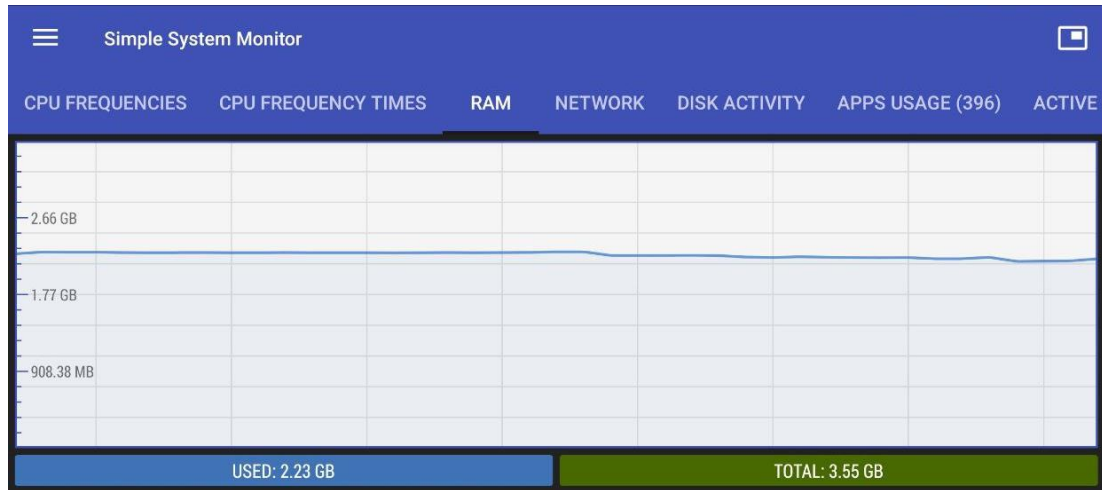


Figura 11. Ejecución de pruebas de rendimiento uso de memoria RAM.

Ejecución de pruebas de rendimiento de uso de la batería con la herramienta Simple System Monitor durante el uso de la aplicación en primer plano (10, 20, 30 y 60) minutos y en segundo plano (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 y 24) horas, una vez generado el reporte los datos fueron almacenados para su posterior análisis.

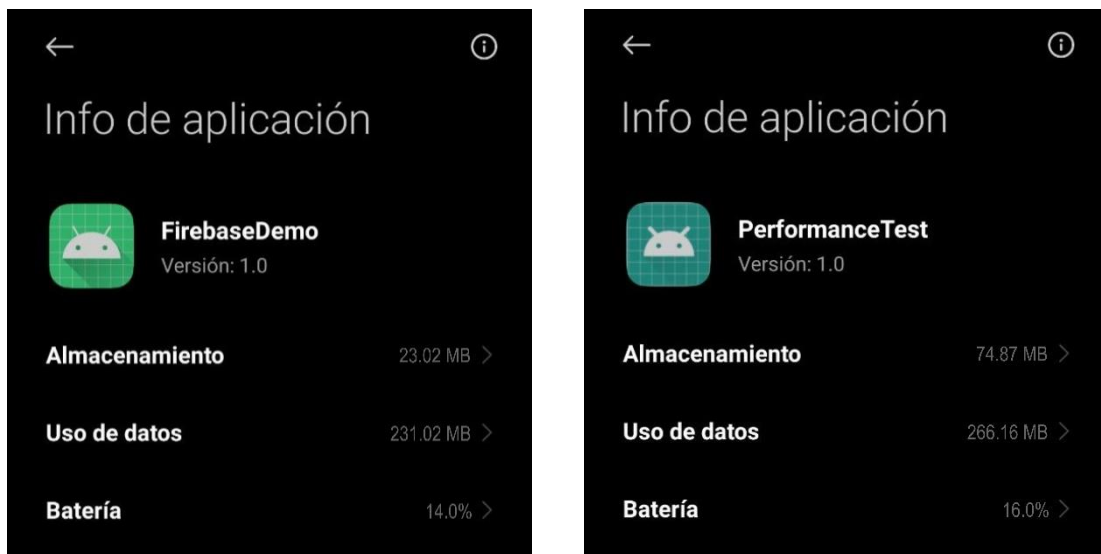


Figura 12. Ejecución de pruebas de rendimiento uso de la batería.

Ejecución de pruebas de rendimiento del tiempo de respuesta en las aplicaciones móviles utilizando la herramienta Simple System Monitor durante la instalación, inicio y ejecución de las aplicaciones, una vez generado el reporte los datos fueron almacenados para su posterior análisis.

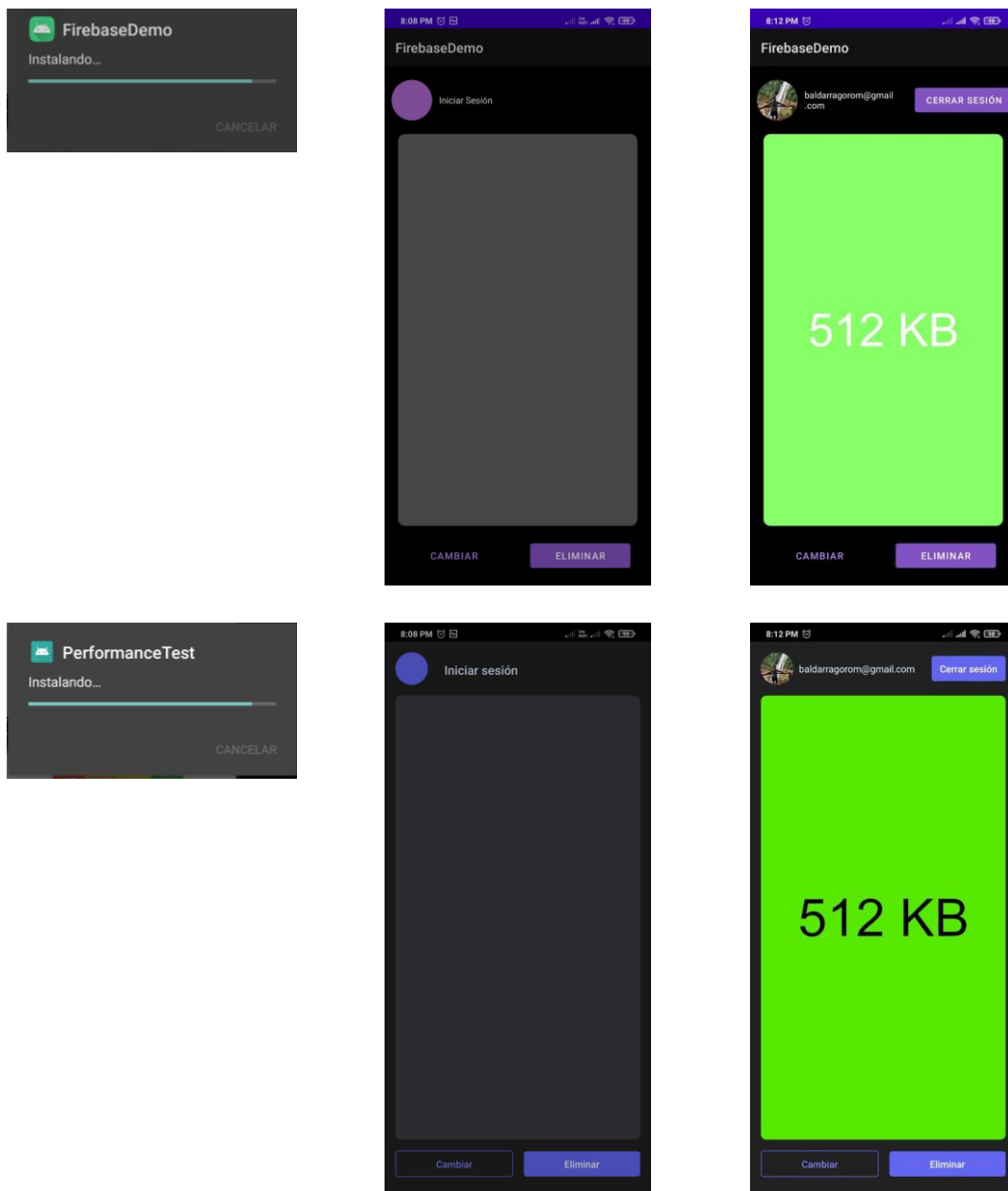


Figura 13. Ejecución de pruebas de tiempo de respuesta.

Proceso 05: Realizar pruebas de conexión a la red

Ejecución de pruebas de rendimiento de conexión a la red en las aplicaciones móviles utilizando la herramienta Simple System Monitor durante de la transferencia de paquetes de diferentes tamaños (512 KB, 1 MB, 5MB, 10MB Y 20MB), una vez generado el reporte los datos fueron almacenados para su posterior análisis.

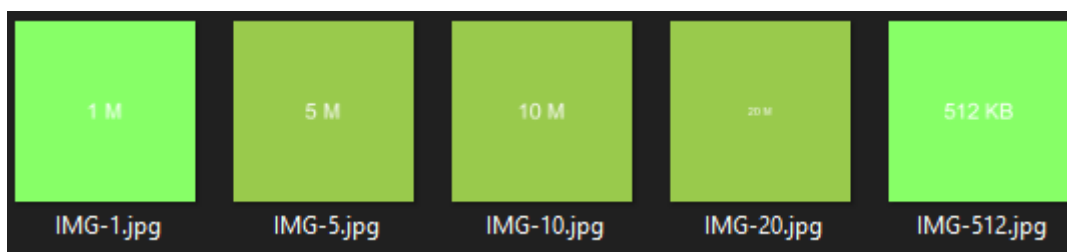


Figura 14. Archivos preparados para la ejecución de la prueba de conexión a la red.

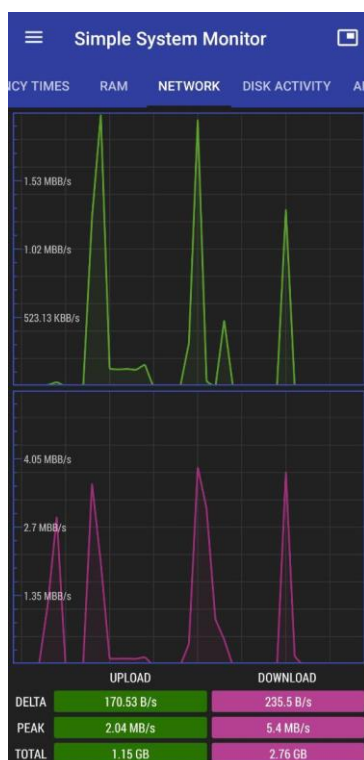


Figura 15. Ejecución de prueba de rendimiento de conexión a la red.

Proceso 06: Realizar pruebas de carencia de seguridad

Ejecución de pruebas de carencia de seguridad utilizando la herramienta Immuniweb para hallar el número de vulnerabilidades en cada una de las



UNIVERSIDAD CÉSAR VALLEJO

**FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

Declaratoria de Originalidad de los Autores

Nosotros, BALDARRAGO PILCO ROMARIO ALBERTO, MENDOZA HUAMAN ANDERSON CHRISTOFER estudiantes de la FACULTAD DE INGENIERÍA Y ARQUITECTURA de la escuela profesional de INGENIERÍA DE SISTEMAS de la UNIVERSIDAD CÉSAR VALLEJO SAC - LIMA ESTE, declaramos bajo juramento que todos los datos e información que acompañan la Tesis titulada: "METODOLOGÍA PARA LA EVALUACIÓN DEL RENDIMIENTO DE APLICACIONES MÓVILES", es de nuestra autoría, por lo tanto, declaramos que la Tesis:

1. No ha sido plagiada ni total, ni parcialmente.
2. Hemos mencionado todas las fuentes empleadas, identificando correctamente toda cita textual o de paráfrasis proveniente de otras fuentes.
3. No ha sido publicada, ni presentada anteriormente para la obtención de otro grado académico o título profesional.
4. Los datos presentados en los resultados no han sido falseados, ni duplicados, ni copiados.

En tal sentido asumimos la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de la información aportada, por lo cual nos sometemos a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

Nombres y Apellidos	Firma
BALDARRAGO PILCO ROMARIO ALBERTO DNI: 71847067 ORCID 0000-0002-4326-0799	Firmado digitalmente por: RBALDARRAGO el 11-12-2021 08:25:16
MENDOZA HUAMAN ANDERSON CHRISTOFER DNI: 74911474 ORCID 0000-0001-8777-0427	Firmado digitalmente por: AMENDOZAHU el 11-12-2021 08:28:49

Código documento Trilce: INV - 0478727