



**FACULTAD DE INGENIERÍA ARQUITECTURA Y  
URBANISMO**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**TESIS**

**ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS  
DE DESARROLLO PARA APLICACIONES MÓVILES  
HÍBRIDAS**

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

**Autor:**

**Bach. De la Cruz Ceron Roberto Carlos**

**<https://orcid.org/0000-0002-2680-2955>**

**Asesor:**

**Mg. Atalaya Urrutia Carlos William**

**<https://orcid.org/0000-0002-2761-4868>**

**Línea de Investigación:**

**Infraestructura, Tecnología y Medio Ambiente**

**Lima– Perú**

**2021**

**ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS DE DESARROLLO PARA  
APLICACIONES MÓVILES HÍBRIDAS**

**JURADO CALIFICADOR**

---

Mg. Atalaya Urrutia Carlos William  
Asesor Especialista

---

MSc. Guerrero Millones Ana María  
Asesora Metodológica

---

Mg. Bances Saavedra David Enrique  
Presidente

---

Mg. Chirinos Mundaca Carlos Alberto  
Secretario

---

Mg. Diaz Vidarte Miguel Orlando  
Vocal

### **DEDICATORIA:**

Este trabajo va dedicado con todo mi cariño a mis padres, quienes me dieron la educación y formación correcta para convertirme en un hombre de bien.

A mi abuelita Irma quien no descansó hasta lograr convencerme de venir a probar suerte en mi primer trabajo el cual fue el primer peldaño para iniciar mi carrera profesional.

A toda mi familia que siempre fue y será mi motor para seguir creciendo y desarrollándome como persona y profesional.

Finalmente, a todos los que creyeron en mí y apostaron por mi cuando ni yo mismo creía que podía llegar tan lejos.

## **AGRADECIMIENTO:**

En primer lugar, a Dios por haberme dado la vida y la salud que me ha permitido llegar hasta aquí y terminar la carrera profesional.

A todas las personas que de una u otra manera contribuyeron a materializar mis objetivos profesionales.

A la MSc. Ana María Guerrero Millones, por su experiencia científica, sus valiosas críticas e incondicional apoyo en la revisión y esclarecimiento de las diferentes etapas para la concreción de la Tesis.

Mil gracias.

## ÍNDICE

I. INTRODUCCIÓN .....	13
1.1. Realidad Problemática .....	13
1.2. Trabajos previos.....	19
1.3. Teorías relacionadas al tema. ....	24
1.1.1. Frameworks .....	24
1.1.2. Aplicaciones Móviles .....	27
1.1.3. Normativa técnica .....	32
1.1.4. Gestión de Riesgos .....	35
1.1.5. Modelo de evaluación.....	36
1.1.6. Estado del Arte .....	37
1.1.7. Definición de términos .....	38
1.4. Formulación del Problema. ....	40
1.5. Justificación e importancia del estudio. ....	41
1.6. Hipótesis. ....	42
1.7. Objetivos. ....	42
1.1.8. Objetivo General.....	42
1.1.9. Objetivos específicos.....	43
II. MATERIAL Y MÉTODOS .....	44
2.1. Tipo y diseño de investigación. ....	44
2.2. Población, muestra y muestreo.....	44
2.2.1. Población.....	44
2.2.2. Muestra.....	44
2.3. Variables y Operacionalización .....	44
2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad. . .....	46
2.4.1. Confiabilidad de instrumentos .....	47
2.4.2. Validación de instrumentos.....	49
2.5. Procedimiento de análisis de datos.....	49
2.5.1. Métodos de análisis de datos .....	52
2.6. Criterios éticos .....	52
2.6.1. Ambiental.....	53
2.6.2. Objetividad.....	53
2.6.3. Originalidad .....	53
2.6.4. Veracidad .....	53
2.7. Criterios de Rigor Científico .....	53

III. RESULTADOS .....	54
4.1. Presentación de resultados .....	54
4.1.1. Seleccionar los frameworks para realizar la comparativa .....	54
4.1.2. Desarrollar una aplicación como caso de estudio para la comparación. ....	55
4.1.3. Realizar pruebas a las aplicaciones desarrolladas con base en los frameworks. ....	60
4.2. Discusión de resultados .....	67
4.3. Aporte práctico .....	68
V. CONCLUSIONES Y RECOMENDACIONES .....	106
6.1. Conclusiones .....	106
6.2. Recomendaciones .....	107

## ÍNDICE DE TABLAS

Tabla 1.	Matriz de operacionalización de variables .....	45
Tabla 2.	Estadísticas de escala para cuestionario 01 .....	48
Tabla 3.	Estadísticas de fiabilidad para cuestionario 01 .....	48
Tabla 4.	Estadísticas de escala para cuestionario 02.....	48
Tabla 5.	Estadísticas de fiabilidad para cuestionario 02 .....	49
Tabla 6.	Procedimiento de análisis de datos .....	50
Tabla 7.	Matriz de objetivos específicos e instrumentos de recolección de información.....	51
Tabla 8.	Horas de esfuerzo por Framework para creación del CRUD .....	56
Tabla 9.	Dificultad para obtener documentación e integración al proyecto.....	56
Tabla 10.	Número total de horas de esfuerzo para desarrollar CRUD .....	57
Tabla 11.	Nivel de dificultad de la instalación por Framework .....	58
Tabla 12.	Consumo de recursos de PC por Framework.....	58
Tabla 13.	Consumo de recursos de PC por aplicación desarrollada - Android..	61
Tabla 14.	Consumo de recursos de PC por aplicación desarrollada - IOS.....	63
Tabla 15.	Tiempo de respuesta por operación CRUD - Android .....	65
Tabla 16.	Tiempo de respuesta por operación CRUD - IOS.....	65
Tabla 17.	Categoría del Top G2 CROWD .....	70
Tabla 18.	Criterios del Top G2 CROWD.....	71
Tabla 19.	Diccionario de datos .....	77
Tabla 20.	Consumo de memoria RAM por cada una de las operaciones CRUD dentro del plan de pruebas realizado - Ionic Framework - Android .....	86
Tabla 21.	Consumo de memoria RAM en promedio con base en el total de pruebas realizadas - Ionic Framework - Android.....	87
Tabla 22.	Consumo de memoria RAM por cada una de las operaciones CRUD dentro del plan de pruebas realizado - Ionic Framework - IOS .....	88
Tabla 23.	Consumo de memoria RAM en promedio con base en el total de pruebas realizadas - Ionic Framework – IOS .....	89
Tabla 24.	Consumo de memoria RAM por cada una de las operaciones CRUD dentro del plan de pruebas – Apache Cordova - Android.....	89
Tabla 25.	Consumo de memoria RAM en promedio con base en el total de pruebas realizadas - Apache Cordova - Android.....	90
Tabla 26.	Consumo de memoria RAM por cada una de las operaciones CRUD dentro del plan de pruebas – Apache Cordova - IOS.....	91
Tabla 27.	Consumo de memoria RAM en promedio con base en el total de	

pruebas realizadas - Apache Cordova - IOS.....	92
Tabla 28. Porcentaje de CPU con que consumen las operaciones CRUD, para cada una de las tareas dentro del plan de pruebas – Ionic Framework - Android.....	92
Tabla 29. Porcentaje de consumo de CPU en promedio para cada una de las operaciones CRUD - Ionic Framework - Android .....	94
Tabla 30. Porcentaje de CPU con que consumen las operaciones CRUD, para cada una de las tareas dentro del plan de pruebas – Ionic Framework - IOS.....	94
Tabla 31. Porcentaje de consumo de CPU en promedio para cada una de las operaciones CRUD - Ionic Framework - IOS.....	95
Tabla 32. Porcentaje de CPU con que consumen las operaciones CRUD, para cada una de las tareas dentro del plan de pruebas – Apache Cordova - Android.....	96
Tabla 33. Porcentaje de consumo de CPU en promedio para cada una de las operaciones CRUD – Apache Cordova - Android .....	97
Tabla 34. Porcentaje de CPU con que consumen las operaciones CRUD, para cada una de las tareas dentro del plan de pruebas – Apache Cordova - IOS.....	97
Tabla 35. Porcentaje de consumo de CPU en promedio para cada una de las operaciones CRUD – Apache Cordova - IOS.....	98
Tabla 36. Tiempos de respuesta para cada una de las operaciones CRUD dentro del plan de pruebas – Ionic Framework - Android.....	99
Tabla 37. Tiempos de respuesta en promedio cada una de las operaciones CRUD – Ionic Framework - Android .....	100
Tabla 38. Tiempos de respuesta para cada una de las operaciones CRUD dentro del plan de pruebas – Ionic Framework - IOS .....	100
Tabla 39. Tiempos de respuesta en promedio cada una de las operaciones CRUD – Ionic Framework - IOS .....	102
Tabla 40. Tiempos de respuesta para cada una de las operaciones CRUD dentro del plan de pruebas – Apache Cordova - Android.....	102
Tabla 41. Tiempos de respuesta en promedio cada una de las operaciones CRUD – Apache Cordova – Android.....	103
Tabla 42. Tiempos de respuesta para cada una de las operaciones CRUD dentro del plan de pruebas – Apache Cordova - IOS.....	103
Tabla 43. Tiempos de respuesta en promedio cada una de las operaciones CRUD – Apache Cordova – IOS.....	105

## ÍNDICE DE FIGURAS

<i>Figura 1.</i> Ilustra el porcentaje de empresas que utilizan diversas tecnologías de la información y la comunicación. Encuesta económica anual 2019 - INEI .....	17
<i>Figura 2.</i> Porcentaje de empresas con página web. Encuesta económica anual 2019 - INEI .....	17
<i>Figura 3.</i> Según el comercio electrónico, el porcentaje de empresas. Encuesta económica anual 2019 - INEI .....	18
<i>Figura 4.</i> Por sectores empresariales, el porcentaje de empresas que gastan en investigación y tecnología. Encuesta económica anual 2019 - INEI .....	18
<i>Figura 5.</i> Frameworks móviles híbridos utilizados por desarrolladores de software durante los años 2019 a 2021. Fuente: (Varun Bhagat, Hybrid Mobile App Frameworks You Can Opt for in 2021 & Beyond, 2021) .....	54
<i>Figura 6.</i> Consumo de memoria RAM por cada uno de los frameworks. Fuente: Elaboración propia. ....	59
<i>Figura 7.</i> Consumo de porcentaje de CPU por Framework. Fuente: Elaboración Propia. ....	59
<i>Figura 8.</i> Consumo de Memoria RAM por aplicación desarrollada. Fuente: Elaboración propia .....	62
<i>Figura 9.</i> Consumo de porcentaje de CPU por aplicación desarrollada - Android. Fuente: Elaboración propia. ....	62
<i>Figura 10.</i> Consumo de Memoria RAM por aplicación desarrollada - IOS. Fuente: Elaboración propia .....	63
<i>Figura 11.</i> Consumo de porcentaje de CPU por aplicación desarrollada. Fuente: Elaboración propia. ....	64
<i>Figura 12.</i> Tiempos de respuestas desglosado por cada una de las operaciones CRUD. Fuente: Elaboración propia. ....	65
<i>Figura 13.</i> Tiempos de respuestas desglosado por cada una de las operaciones CRUD. Fuente: Elaboración propia. ....	66
<i>Figura 14.</i> Ranking de los frameworks más usados actualmente. Fuente: G2 CROWD (2021) .....	70
<i>Figura 15.</i> Arquitectura propia del framework Apache Cordova. Fuente Apache Cordova 2021.....	73
<i>Figura 16.</i> Arquitectura propia que provee Ionic Framework. Fuente: Ionic Framework 2021. ....	74
<i>Figura 17.</i> Arquitectura propuesta para el desarrollo de la aplicación móvil. Fuente: Elaboración propia. ....	75
<i>Figura 18.</i> Tabla de usuarios desarrollada por el H2. Fuente: Elaboración	

propia.	76
<i>Figura 19.</i> Consola de administración para el motor de base de datos H2. Fuente: Elaboración propia. ....	77
<i>Figura 20.</i> Captura de pantalla que muestra el entorno de desarrollo para los servicios web con Java Sprinboot. Fuente: Elaboración propia. ....	78
<i>Figura 21.</i> Captura de pantalla que muestra el entorno de desarrollo a nivel de frontend. Fuente: Elaboración propia. ....	79
<i>Figura 22.</i> Interfaz gráfica para agregar crear nuevos usuarios. Fuente: Elaboración propia. ....	80
<i>Figura 23.</i> Interfaz gráfica que muestra el proceso de creación de usuarios exitoso. Fuente: Elaboración propia. ....	81
<i>Figura 24.</i> Interfaz gráfica que lista lo usuarios creados por la aplicación. Fuente: Elaboración propia. ....	82
<i>Figura 25.</i> Interfaz gráfica que muestra edición de usuarios. Fuente: Elaboración propia.	83
<i>Figura 26.</i> Interfaz gráfica que muestra el mensaje satisfactorio luego de realizar la edición de un usuario. Fuente: Elaboración propia.....	84
<i>Figura 27.</i> Interfaz gráfica que muestra el mensaje de confirmación antes de realizar la eliminación del usuario seleccionado. Fuente: Elaboración propia.....	85

## RESUMEN

El uso de frameworks de desarrollo cada vez es más común en equipos de desarrollo de software durante los últimos años, el uso adecuado de un framework específico provee una estructura definida para el proyecto y mejora notablemente los tiempos de desarrollo y entrega de un producto de software, pero, La elección de un marco de trabajo suele suponer un reto para el equipo de desarrollo a la hora de determinar cuál es el que mejor se adapta a las necesidades del proyecto. Por ello, esta investigación aplicada y de diseño cuasi-experimental con enfoque cuantitativo pretende abordar el problema de cómo elegir el framework que mejor se adapte a las necesidades de un equipo de desarrollo a la hora de implementar una aplicación móvil híbrida que tiene funcionalidades básicas como un CRUD de usuarios (crear, listar, actualizar y eliminar). Como resultados se obtienen el consumo de recursos por cada framework así mismo por la aplicación desarrollada por cada uno y los tiempos de respuesta al realizar las operaciones definidas anteriormente. Se concluye que ninguno de los frameworks es mejor que el otro ya que cada uno posee sus atributos particulares, pero ambos frameworks contienen librerías de Apache Cordova por lo tanto el trabajo con cada uno de ellos es similar y supone gran facilidad al momento de la integración con cualquier proyecto desarrollado con JavaScript, HTML y CSS.

### **Palabras clave:**

Frameworks, aplicaciones móviles híbridas, desarrollo de software, análisis comparativo, aplicaciones móviles.

## **ABSTRACT**

The use of development frameworks has become more and more common in software development teams in recent years, the use of a specific framework provides a defined structure for the project and significantly improves the development and delivery times of a software product but Choosing a certain framework is often a problem for the development team when it comes to selecting the one that best suits the needs of the project, for this reason in this applied research and quasi-experiment design with a quantitative approach it is intended to give solution to the problem of how to choose the framework that best suits the needs of a development team when implementing a hybrid mobile application, for this it is considered convenient to make a comparison between two hybrid mobile application development frameworks through development of a mobile application that has basic functionalities such as a user CRUD (create, read, update and delete). As results, the consumption of resources by each framework is obtained, as well as by the application developed by each one and the response times when performing the operations defined above. It is concluded that none of the frameworks is better than the other since each one has its particular attributes, but both frameworks contain Apache Cordova libraries, therefore the work with each of them is similar and is very easy at the time of integration. with any project developed with Javascript, HTML and CSS.

### **Keywords:**

Frameworks, hybrid mobile applications, software development, comparative analysis, mobile application

## **I. INTRODUCCIÓN**

### **1.1. Realidad Problemática**

Las apps web y las aplicaciones móviles automatizan operaciones que ahora se realizan manualmente en las empresas, lo que permite centralizar la información y la perspectiva de mejorar enormemente la productividad. La empresa "Morales & Asociados", que presta servicios jurídicos de alta calidad y que tiene su sede en Riobamba, utiliza actualmente procesos manuales, que dan lugar a procesos largos. Por ello, el autor propone desarrollar una aplicación web y una aplicación móvil híbrida para automatizar los procesos de control y seguimiento de los juicios. (Haro y Quishpi, 2019).

La informática móvil se describe como un entorno informático que permite a los usuarios acceder a la información digital desde cualquier dispositivo conectado a cualquier red, independientemente de su ubicación geográfica. El desarrollo de aplicaciones para dispositivos móviles introduce obstáculos adicionales debido a las propiedades únicas de la operación. El requisito de abordar un conjunto diverso de normas, protocolos y tecnologías de red, la capacidad limitada de los equipos a pesar del continuo crecimiento en el tiempo impuesto por un mercado muy dinámico, y la presencia de múltiples plataformas de hardware y software son sólo algunas de las dificultades a las que se enfrentan los desarrolladores en este ámbito. (Thomas et al. 2018).

Hoy en día, las app web dominan el mercado de productos de software debido a las numerosas ventajas que ofrecen y su dependencia de Internet para evitar el uso normal del escritorio y desarrollar todo tipo de operaciones. Esto se debe a la estructura autónoma que presentan y la falta de flexibilidad en el cambio constante de la tecnología. Sin embargo, con el crecimiento de este mercado, los problemas relacionados con el desarrollo de estas aplicaciones son cada vez más frecuentes. Como tal, se han generado muchas metodologías que ayudan a guiar el ciclo de vida del software, y que también ayuda a reducir y corregir los errores existentes asociados con cada etapa de desarrollo. Por ello, se realizó un estudio en profundidad para obtener información sobre las metodologías, con el objetivo de comparar frameworks de desarrollo de software web de las mismas

con la bibliografía, análisis y exploración en campo, según diversos autores. Después de la investigación, determinó que el método era el método OOHDM. Este método proporciona una funcionalidad óptima para el área de desarrollo de app web y es el más utilizado por los programadores y desarrolladores de software en el estado de ORO (Molina et al. 2018).

La tecnología avanza rápidamente todos los días, y hace unos años el uso de Internet no era tan necesario como lo es hoy. Si alguien señala que Internet reemplazará al material impreso, será ignorado, pero ahora en realidad se ha vuelto muy competitivo usar Tecnologías Web y todas las empresas van en esa dirección, por eso está progresando. Los desarrolladores de software se dedican principalmente al desarrollo de páginas web, y es fundamental entender qué lenguaje de programación es el óptimo para desarrollar un sitio web, lo cual es ahora un reto. Por tal motivo esta tarea tiene como objetivo utilizar métodos de investigación comparativa para identificar y analizar dos de las herramientas de desarrollo web actualmente en uso, como ASP.NET y PHP. Se presentan las fortalezas y debilidades de cada uno, investigadas por diferentes autores, y se le da al lector una idea general de lo que él puede lograr. Esto es fundamental para identificar la herramienta más adecuada para los desarrolladores de apps web. Se consideran los aspectos más relevantes de las dos herramientas para aclarar el a la hora de decidir cuál elegir. Ya sea funcional o administrativo, es importante conocer las opciones que ofrece el lenguaje de programación para elegir. Al elegir un lenguaje de programación, debe tener en cuenta todas las características y definiciones proporcionadas. Esto permite convertir el software en un producto de alta calidad que agrega valor a las operaciones de la empresa (Sierra y Espinoza, 2018).

JavaScript ha sido y sigue siendo uno de los lenguajes de programación más utilizados en el desarrollo web, lo que ha provocado el surgimiento de varios frameworks de JavaScript, lo que dificulta su selección. La investigación del autor tiene en cuenta la usabilidad a la hora de considerar la adopción del marco de trabajo de JavaScript y se propone evaluar la usabilidad de tres destacados marcos de desarrollo: Backbone, Angular y Ember. La evaluación de la usabilidad se llevó a cabo utilizando el enfoque de las pruebas de usabilidad y

se concentró en las siguientes características: facilidad de aprendizaje, tasa de errores y satisfacción. Los resultados indican que Angular es el framework más sencillo de utilizar y es el que el autor recomienda, seguido de Ember y Backbone. (Navajas y Pamplona, 2018).

La selección adecuada de metodologías de desarrollo supone un problema en la actualidad, elegir uno que se adapte a las necesidades del proyecto ayudan a administrar los planes del proyecto de desarrollo de una manera ágil y eficiente. Esto permite planificar la relación entre el cliente y el equipo de trabajo para cada tipo de proyecto de manera independiente a su complejidad y tamaño, así mismo la comparación de herramientas de desarrollo y sus opciones adecuadas depende del entorno de desarrollo en el que se encuentre basado. Se tiene en cuenta los lenguajes de programación, el rendimiento y el rendimiento del nivel de desarrollador tomando en cuenta los recursos disponibles para empresas o desarrolladores. Se concluye que el framework de desarrollo de Laravel se seleccionó en base a una comparación entre el framework propuesto y el framework evaluado. Esto se debe a que el aspecto de desarrollo es superior a otros frameworks y es adecuado para las aplicaciones de web (Francia, 2019).

Específicamente, los distribuidores de productos deben crear pedidos, recibir recomendaciones de inventario de productos en tiempo real y ser más competitivos con empresas. La entrada de pedidos ha tenido una enorme influencia en el campo del comercio móvil, lo que ha dado lugar al desarrollo de una serie de tácticas o sistemas. Utilizando esta tecnología, las empresas del primer mundo pueden adelantarse a la competencia reduciendo los retrasos en los pedidos a proveedores y el registro de productos, así como las líneas de error en las entregas. La cuestión es que en la actualidad sólo las empresas estratégicamente competitivas de Perú tienen acceso a este instrumento para agilizar el procedimiento de registro de pedidos. Estas grandes empresas, que ahora son líderes en el mercado, son conscientes del papel fundamental y del uso de los dispositivos móviles en este proceso. El registro de pedidos a través de una aplicación móvil permite realizar pedidos, examinar artículos y enviarlos a un servidor central en línea. Por otro lado, las pequeñas y medianas empresas (PyMEs) desconocen la ventaja competitiva que las acompaña porque carecen

de los recursos financieros o del conocimiento para operar el sistema móvil (Castillo y Dávila, 2020).

Las aplicaciones móviles han modificado la forma de interactuar de las personas. Debido a este cambio en el mundo tecnológico y el avance de la ciencia, supone un problema para los ingenieros a la hora de elegir un marco de trabajo para desarrollar un proyecto. Con la existencia de diferentes frameworks de desarrollo de aplicaciones para todo tipo de actividades, esto también obliga a los ingenieros a desarrollar continuamente herramientas de software. Del mismo modo, Internet como herramienta facilita interacciones entre empresas y clientes. Gracias a esta tecnología, las personas pueden ahorrar tiempo al obtener información y compartirla con personas (Palomino, 2018).

Es posible proporcionar información que permita el desarrollo de apps móviles para que la ciudadanía pueda encontrar e identificar PyMEs. En el contexto actual, facilita el desarrollo de este tipo de solución que se puede aplicar en todas las etapas del proyecto cuando existe un método ágil involucrando todas las fases como análisis, diseño, desarrollo y prueba del producto final. En el área del desarrollo móvil, existen tecnologías que le permiten construir esta solución en múltiples plataformas mediante el uso de herramientas híbridas, en lugar de enfocarse en una sola plataforma, como Android o iOS. Esta tecnología puede manejar una única codificación, lo que reduce el tiempo de desarrollo. Esta es una ventaja sobre el mercado actual. Con la aplicación, puede llegar a tantos usuarios como sea posible, lo cual es bueno para la aplicación y ayuda a su pequeña empresa a llegar a más clientes (Montes, 2020).

La Encuesta Económica Anual del Instituto Nacional de Estadística e Informática examina las grandes, medianas y pequeñas empresas que desarrollaron actividad económica en 2019. En ese periodo de tiempo se registraron 100 mil 627 empresas, de las cuales el 94,1 por ciento utilizó la telefonía móvil, el 93,8 por ciento el ordenador, el 92,2 por ciento internet, el 84,6 por ciento la telefonía fija, el 17,4 por ciento la intranet, el 15,1 por ciento la PDA (Asistente Digital Personal) / Tablet y el 7,4 por ciento la extranet.

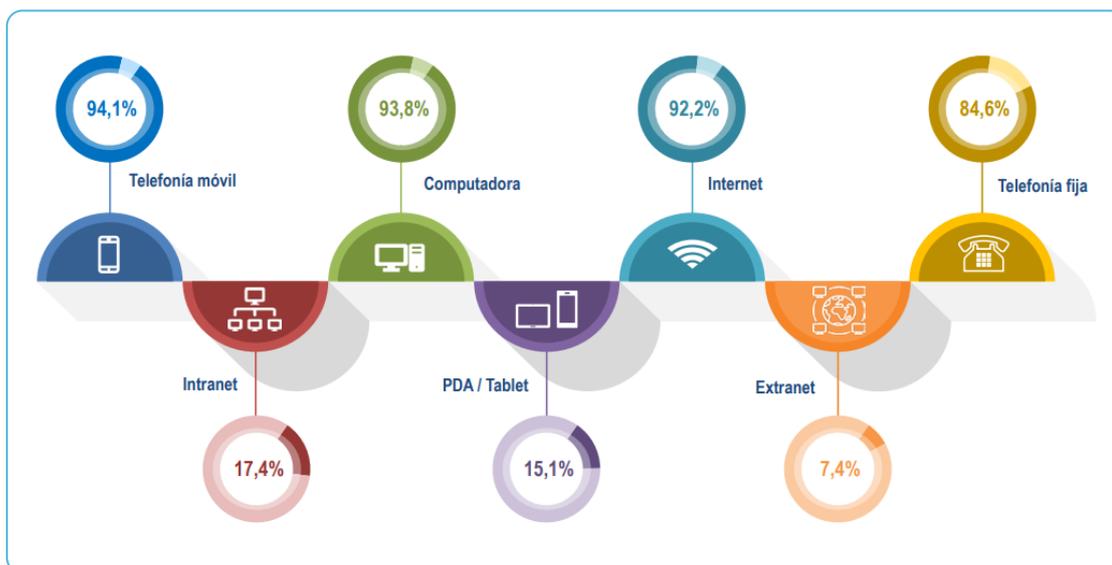


Figura 1. Ilustra el porcentaje de empresas que utilizan diversas tecnologías de la información y la comunicación. Encuesta económica anual 2019 - INEI

El 29,2 por ciento de las empresas dispone de una página web que le permite ofrecer de forma continuada sus bienes y/o servicios, mientras que el 70,8 por ciento no la tiene.

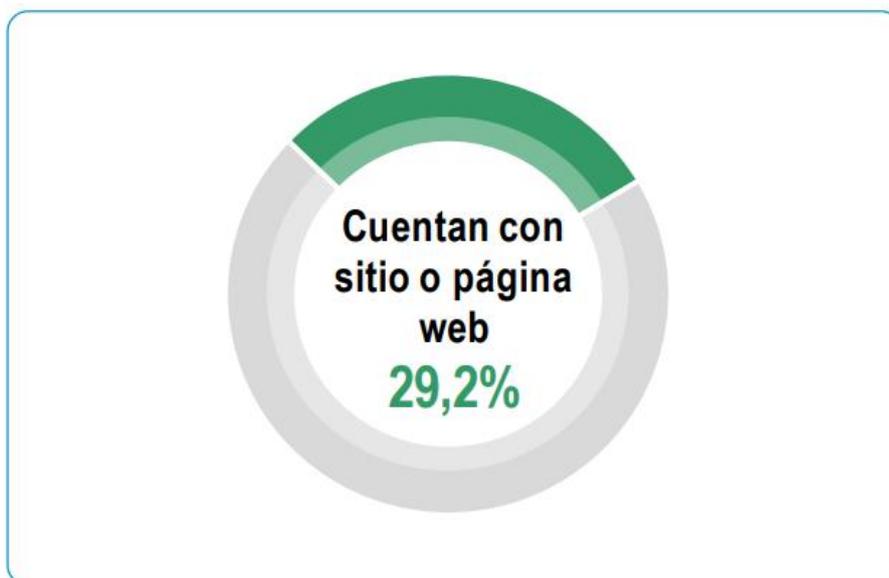
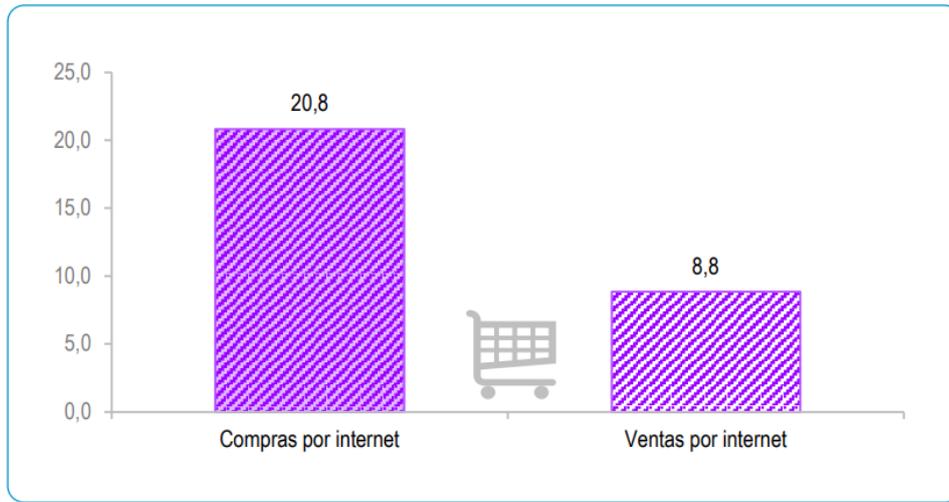


Figura 2. Porcentaje de empresas con página web. Encuesta económica anual 2019 - INEI

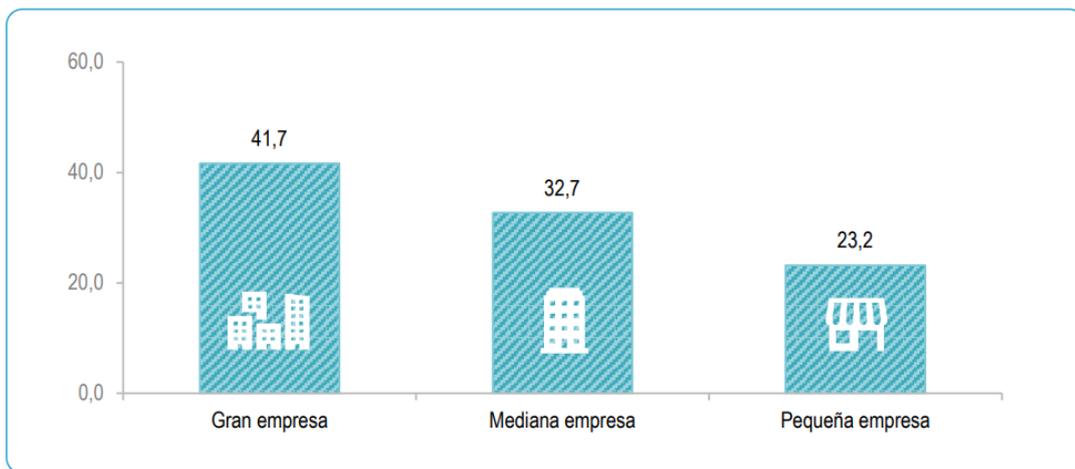
El 20,8 por ciento de los negocios compraron artículos y/o servicios en línea, mientras que el 8,8 por ciento de los negocios ofrecieron productos y/o servicios en

línea.



*Figura 3.* Según el comercio electrónico, el porcentaje de empresas. Encuesta económica anual 2019 - INEI

El 41,7% de las grandes empresas, el 32,7% de las organizaciones medianas y el 23,2% de las pequeñas empresas invirtieron en ciencia y tecnología, lo que demuestra una clara correlación entre la inversión en ciencia y tecnología y el sector empresarial.



*Figura 4.* Por sectores empresariales, el porcentaje de empresas que gastan en investigación y tecnología. Encuesta económica anual 2019 - INEI

Actualmente dentro de empresas dedicadas específicamente de desarrollo de software, así como otras empresas que desarrollan su propio software, los equipos de desarrollo al momento de iniciar un proyecto cuestionan qué tecnologías podrían utilizarse para el desarrollo del producto de software, desde la infraestructura, base de datos, lenguajes de programación, frameworks de desarrollo, librerías de terceros, etc. Cada equipo de desarrollo evalúa según diversos criterios las tecnologías a utilizar, se consideran por ejemplo: si se tiene software implementado se puede utilizar las tecnologías con las que se han desarrollado el software anterior (reutilizar licencias de base de datos, servidores, frameworks, etc.), de no ser el caso se puede considerar el costo beneficio de elegir las tecnologías, el conocimiento técnico por parte del equipo así como la curva de aprendizaje y el alcance del software a desarrollar.

En el mercado se cuentan con diversas tecnologías para implementar proyectos de desarrollo de software, pero los equipos siempre o casi siempre cuentan con ciertas dificultades a la hora de elegir un framework apropiado para desarrollar aplicaciones.

## **1.2. Trabajos previos**

Robin Nunkesser, (2018), realizó la investigación *Beyond Web/Native/Hybrid: A New Taxonomy for Mobile App Development*, en Alemania. En la actualidad en el mundo de las aplicaciones móviles los sistemas operativos que dominan el mercado son IOS y Android, sin embargo, éstas no cuentan con un framework compatible en común, en cambio hay diferentes enfoques para el desarrollo de aplicaciones multiplataforma. Por esta razón el autor en su trabajo propone una nueva taxonomía sobre la base en la diferencia entre herramientas y lenguajes de programación utilizados donde se propone seis categorías diferentes: aplicaciones endémicas, aplicaciones web, aplicaciones web híbridas, aplicaciones puente híbridas, aplicaciones en el idioma del sistema y aplicaciones escritas en un idioma extranjero, así mismo se puede evidenciar una breve descripción entre sus ventajas y desventajas.

Valerio Panzica La Manna y Frank Pasveer, (2018), en su investigación *Towards a Framework for Proximity-Based Hybrid Mobile Applications* en el país de

Suecia. A pesar de la creciente disponibilidad de dispositivos y tecnologías de IoT, la interacción del usuario actual requiere que los usuarios descarguen aplicaciones móviles específicas ad-hoc, cada una de las cuales puede monitorear y controlar de forma remota un número limitado de dispositivos de IoT. En un futuro cercano, este enfoque no se escalará con miles de millones de dispositivos IoT implementados, ya que no se puede esperar que los usuarios descarguen diferentes aplicaciones para cada ubicación que visitan o con cada dispositivo con el que interactúan. Para superar esta limitación, el autor presenta en su investigación la visión, los riesgos y las oportunidades de desarrollar e implementar futuros servicios de proximidad híbridos como un cambio de paradigma para la interacción intuitiva entre los usuarios y el entorno de IoT circundante. Los servicios de proximidad híbridos son aplicaciones móviles centradas en el usuario diseñadas y configuradas para ubicaciones específicas y que se instalan automáticamente en los dispositivos móviles del usuario cuando el usuario está cerca y sin la necesidad de una descarga explícita. Un entorno de servicios de proximidad puede proporcionar directamente al usuario el servicio adecuado para el lugar adecuado e instruir automáticamente al dispositivo del usuario para que interactúe con los dispositivos IoT circundantes.

Yahui Yang, Youbing Zhang, Pengtao Xia, Bo Li y Zezhi Ren, (2017) en la investigación *Mobile Terminal Development Plan of Cross-Platform Mobile Application Service Platform Based on Ionic and Cordova* desarrollada en China. La situación actual implica alta demanda de aplicaciones móviles junto con la tendencia de la actual tecnología móvil, en el estudio se analizan tres tipos de aplicaciones móviles en función a los procesos de una empresa de automotriz, utilizando Ionic Framework en fusión con AngularJS para la interfaz de usuario y utilizando el motor híbrido Cordova para la integración multiplataforma híbrida de dispositivos móviles. Estas herramientas utilizan WebStorm para el desarrollo de páginas front-end. Los resultados del estudio indican que la estrategia tecnológica del plan de desarrollo es factible y proporciona una solución de terminal móvil para que las empresas integren y unifiquen la experiencia del usuario, así como mejora la capacidad de la empresa para poder controlar la información móvil, de esta manera sienta una base sólida para que las empresas

logren información móvil de manera rápida y eficiente.

Carlos Manso Pinto y Carlos Coutinho, (2018) en su investigación *From Native to Cross-platform Hybrid Development* realizada en Portugal. Las tendencias actuales hacia la globalización, junto con el crecimiento exponencial en el número de dispositivos móviles, han llevado a los mercados a un auge en la demanda de desarrollo de apps móviles. Además, con la expansión y heterogeneidad de los dispositivos y plataformas móviles, las empresas de software deben buscar formas más rápidas y económicas de desarrollar aplicaciones que puedan extenderse a tantos dispositivos como sea posible para conquistar el mercado. Actualmente, los sistemas operativos Android e iOS comparten y dominan aproximadamente el mercado móvil, con expresiones tímidas de otros competidores. Cada uno de estos sistemas operativos móviles se ha desarrollado utilizando sus propios lenguajes y estrategia, y ofrecen SDK para desarrollar aplicaciones utilizando sus propias bibliotecas, conocidas como aplicaciones nativas. Por otro lado, la evolución de HTML5, CSS y JavaScript ha creado alternativas genéricas para la construcción de aplicaciones móviles que funcionan en dispositivos en todos los sistemas operativos, sin embargo, faltan algunas características para acceder a todo el potencial del dispositivo. Junto con ellos llegaron los nuevos marcos de desarrollo híbridos multiplataforma, tratando de aprovechar lo mejor de ambos mundos. Plataformas nativas versus plataformas híbridas la tendencia en el desarrollo de apps móviles utilizando plataformas híbridas y su ventaja.

Ivano Malavolta, (2016), en su estudio *Web-Based Hybrid Mobile Apps: State of the Practice and Research Opportunities* desarrollado en Estados Unidos. En la actualidad, se descargan y utilizan millones de aplicaciones móviles en todo el mundo. Las aplicaciones móviles se venden a través de varias tiendas de aplicaciones como Google Play Store, Apple App Store, Windows Phone Store. Uno de los desafíos más fascinantes en el desarrollo de aplicaciones móviles es su fragmentación en relación con las plataformas móviles (por ejemplo, Android, Apple iOS, Windows Phone). Recientemente, empresas como IBM y Adobe, así como una creciente comunidad de desarrolladores, han abogado por el desarrollo de aplicaciones móviles híbridas basadas en la web como una posible

solución a la fragmentación de la plataforma móvil. Las aplicaciones móviles híbridas basadas en la web son consistentes en todas las plataformas y se basan en estándares web como HTML5, CSS3, JavaScript. Esta investigación proporciona una descripción general del estado actual de las soluciones, tecnologías y oportunidades de investigación relacionadas con el desarrollo de aplicaciones móviles híbridas basadas en la web. Apache Cordova se presenta como una de las posibles tecnologías para desarrollar aplicaciones híbridas. Los resultados de dos estudios empíricos sobre aplicaciones móviles híbridas reales se presentan como una instantánea actual del estado de la práctica.

Stefan Bosnic, Ištvan Papp y Sebastián Novak, (2016), en su trabajo de investigación denominada *The development of hybrid mobile applications with Apache Cordova* realizada en Serbia. Las aplicaciones móviles nativas son aplicaciones desarrolladas utilizando el SDK y el lenguaje de programación específico de la plataforma móvil. La principal limitación de estas aplicaciones móviles es la imposibilidad de transferir las aplicaciones a otra plataforma sin escribir la aplicación desde cero. Esta situación ha dado lugar a nuevos requisitos, como herramientas que permitan la generación de aplicaciones móviles con una única base de código. En su investigación el autor presenta una forma de crear dichas aplicaciones utilizando Apache Cordova. Permite a los desarrolladores implementar aplicaciones en múltiples plataformas, utilizando una única base de código. Las aplicaciones se ejecutan en un contenedor nativo específico de la plataforma. Como principal conclusión el autor indica que el rendimiento de las aplicaciones creadas con tecnología híbrida o multiplataforma es eficiente permite ahorrar tiempo y esfuerzo en los desarrollos.

Bhavit Mathur y Shashank Mouli Satapathy (2019). En el trabajo de investigación de nombre *An Analytical Comparison of Mobile Application Development using Agile Methodologies* realizado en La India. El rápido progreso de las aplicaciones móviles y el entorno altamente competitivo están aumentando el número de proyectos de aplicaciones móviles. Por lo tanto, el objetivo del estudio es encontrar un enfoque adecuado que cumpla con los requisitos de los sistemas de aplicaciones móviles. El autor estudia los procesos ágiles con el fin de cumplir con los requisitos que hacen que el desarrollo de aplicaciones móviles sea fácil

y adecuado a las características de los dispositivos móviles. Se descubrió que los métodos ágiles son los más adecuados para el desarrollo de aplicaciones móviles porque son breves y la flexibilidad de las necesidades reduce el abuso y la pérdida de tiempo. Entre los resultados relevantes se examinó diferentes enfoques ágiles, pero ninguna investigación proporciona razones concretas para adoptar ágil como modelo necesario para el desarrollo de aplicaciones móviles. El autor concluye que, para el desarrollo de aplicaciones móviles, se propone la metodología ágil modificada, que es el mejor enfoque ágil y cumple con las fortalezas de cada enfoque seleccionado para la ingeniería de software móvil y la eliminación de debilidades.

Kumar Vishal y Ajay Shriram Kushwaha, (2018) en su investigación Mobile Application Development Research Based on Xamarin Platform desarrollada en La India. El desarrollo de aplicaciones móviles está en demanda hoy. Existe mucha competencia entre muchos marcos de desarrollo de aplicaciones móviles disponibles en el mercado, incluido el desarrollo de aplicaciones nativas, sitios web móviles con HTML5, frameworks de desarrollo multiplataforma, etc. Xamarin es una de las herramientas de desarrollo de aplicaciones móviles más importantes para desarrollar aplicaciones híbridas. que se puede ejecutar en cualquier plataforma (iOS, Android, Windows (Phone y Store)) con C#. Este documento describe la arquitectura de Xamarin. Forma el marco y presenta un análisis comparativo de Xamarin. Formularios con Android y ReactNative.

Edison Morocho y Estevan Gómez-Torres, (2019) en su investigación denominada Application of ISO 25000 for Comparative Analysis of Cross-Platform Mobile Application Development Tools desarrollada en Ecuador. Es lógico que la comunidad de desarrolladores de aplicaciones móviles elija las plataformas nativas cuando éstas prometen ofrecer un rendimiento superior al tiempo de ejecución de sus productos. Sin embargo, el desarrollo multiplataforma es posible. Este artículo compara la funcionalidad que ofrecen las distintas herramientas soportadas por Apache Cordova para el desarrollo de aplicaciones móviles híbridas de acuerdo con la norma ISO 25000. El objetivo de este estudio es determinar el entorno óptimo para el desarrollo de aplicaciones móviles., a partir de la generación de una misma aplicación móvil,

Con cuatro herramientas: Visual Studio, Ionic, Cocoon y Evo, utilizando diversas circunstancias de ejecución y evaluando criterios comparativos que indiquen dificultades de ejecución, el desarrollador podrá tomar una decisión más informada a la hora de iniciar un proyecto de construcción de una aplicación móvil híbrida.

Peixin Que, Xiao Guo y Maokun Zhu, (2016) en su investigación A Comprehensive Comparison between Hybrid and Native App Paradigms en La India. Las aplicaciones móviles han estado creciendo drásticamente tanto en cantidad como en calidad en los últimos años, por lo que surgieron muchos tipos de aplicaciones, pero en general hay principalmente dos paradigmas de desarrollo de aplicaciones móviles ampliamente utilizados: el modo de aplicación nativa y el modo de aplicación híbrida. El paradigma de la aplicación nativa está basado en la plataforma y la forma original de desarrollo de aplicaciones móviles, el paradigma de la aplicación híbrida es un producto recientemente surgido basado en el marco multiplataforma y el lenguaje HTML5, ambos paradigmas han generado muchas aplicaciones excelentes. Para determinar el método óptimo para desarrollar una aplicación, este artículo presenta el paradigma de la aplicación híbrida y explora las ventajas y desventajas teóricas y empíricas de ambos paradigmas.

### **1.3. Teorías relacionadas al tema.**

En esta sección se discutirán las teorías críticas relevantes para este estudio, con el objetivo de ubicar el tema dentro de un cuerpo de conocimiento más amplio.

#### **1.1.1. Frameworks**

Framework es una frase que se utiliza a menudo en informática general para referirse a una colección de bibliotecas que se utilizan para implementar la estructura estándar de una aplicación, o para formar el shell de la aplicación. Todo ello se hace para ahorrar líneas de código y, por tanto, facilitar el desarrollo mediante la reutilización de código, de modo que el desarrollador no tenga que escribir lo mismo para cada aplicación.

Según Gamma, E. (1995), un marco de trabajo define la arquitectura de la aplicación. Se trata de un método adecuado, ya que el marco se encarga de diseñar la estructura general, su partición en clases y objetos, los roles críticos y la cooperación entre estas clases y objetos.

Según Gamma (1995), un marco de trabajo captura las opciones de diseño que son comunes a su dominio de aplicación; por lo tanto, un marco de trabajo no sólo fomenta la reutilización del código, sino también el diseño.

Un marco de trabajo permite que el desarrollo de una aplicación sea mucho más fácil y rápido, ya que el tiempo no se desperdicia en elementos de diseño innecesarios que a menudo llevan más tiempo que el necesario para construir la propia lógica de la aplicación. Además, las aplicaciones creadas tienen una arquitectura comparable, lo que las hace más sencillas de mantener y más consistentes para los usuarios. Cabe destacar que esto supone una reducción insignificante de la flexibilidad del diseño.

En ocasiones, el desarrollador se encuentra con dificultades a la hora de diseñar una aplicación; esto se hace aún más difícil para el desarrollador de frameworks, porque desarrollar en un framework requiere un cuidado especial, ya que todos esperamos que cuando se lance uno nuevo, sirva para una variedad de aplicaciones con arquitectura y requisitos similares. Es decir, se intenta incorporar una amplia variedad de aplicaciones dentro de un único estándar, lo que puede ser un éxito o un fracaso. Por ello, nos esforzamos en diseñar marcos que sean lo más extensibles y adaptables posible, permitiendo una rápida y fácil corrección de pequeñas modificaciones. Las aplicaciones construidas sobre un marco de trabajo están inextricablemente ligadas a él, por lo que deben adaptarse y crecer al ritmo del marco, ya que un cambio en el marco provocará un cambio notable en la aplicación, dependiendo de lo extremo que sea el cambio.

Según J. Gutiérrez (s.f.), la idea de framework se emplea en una gran variedad de dominios de desarrollo de sistemas de software, no sólo en aplicaciones web. Existen frameworks para el desarrollo de aplicaciones médicas, aplicaciones de visión por ordenador, creación de juegos y casi

cualquier otro tema que se pueda imaginar. En general, la palabra "framework" se refiere a una estructura de software formada por componentes reutilizables y adaptables que se utilizan en la creación de una aplicación. En otras palabras, se puede pensar en un marco como una aplicación general inacabada y flexible a la que se pueden añadir los últimos componentes de una aplicación real.

Los objetivos principales de un marco son acelerar el proceso de desarrollo, reutilizar el código existente y promover buenas técnicas de desarrollo, como el uso de patrones.

Recaman, H., y Guerrero, C. (2009), por su parte, describen un marco de trabajo como una colección de componentes físicos y lógicos organizados de tal manera que pueden ser reutilizados en el diseño y desarrollo de nuevos sistemas de información.

### **Apache Cordova**

"Es un framework que permite el desarrollo de apps móviles multiplataforma combinando los lenguajes de programación HTML5, CSS y JavaScript con instrucciones propias para acceder a las capacidades del teléfono. Con más de 400 mil desarrolladores, es uno de los frameworks más utilizados en la web." 2021) (Apache Cordova).

El motor central de Apache Cordova es completamente de código abierto. Ofrece una interfaz de programación de aplicaciones (API) basada en JavaScript que permite acceder a diversas funciones nativas del sistema operativo. "La API de Apache Cordova se encarga de convertir las instrucciones de JavaScript e interactuar con el sistema operativo nativo. Mientras que las aplicaciones de PhoneGap utilizan HTML, CSS y JavaScript, el resultado final es un archivo genérico que puede convertirse en varias extensiones para ejecutarse en una variedad de sistemas operativos." 2021) (Apache Cordova).

## **Ionic Framework**

Ionic es un framework SDK gratuito y de código abierto para desarrollar aplicaciones móviles híbridas. Ionic es un marco para la construcción de aplicaciones móviles híbridas que se construye sobre Angular y Apache Cordova. Incluye herramientas y servicios para desarrollar aplicaciones móviles híbridas utilizando tecnologías web como CSS, HTML5 y Sass. Utilizando Cordova, las aplicaciones pueden escribirse utilizando estas tecnologías web y luego entregarse a través de tiendas de aplicaciones nativas para su instalación en los dispositivos. Max Lynch, Ben Sperry y Adam Bradley de Drifty Co. desarrollaron Ionic en 2013 y está siendo utilizado por desarrolladores de software de todo el mundo (Ionic, 2021).

### **1.1.2. Aplicaciones Móviles**

Se trata de un tipo de software de aplicación diseñado específicamente para dispositivos móviles como smartphones y tabletas. Las primeras aplicaciones móviles surgieron a principios del siglo XX, y la mayoría de ellas eran videojuegos, calculadoras y editores de tonos de llamada. En 2008, empezaron a aparecer programas más complejos en los dispositivos móviles, aunque la mayoría de ellos estaban diseñados para la comodidad general y contenían funciones como un calendario, la previsión del tiempo o una lista de contactos.

Según (Domínguez, Paredes, & Santacruz, 2014), destaca que las aplicaciones móviles no son sólo programas de escritorio modificados para dispositivos de pantalla pequeña, sino que son aplicaciones distintas para fines distintos: La capacidad de conectarse con un programa desde cualquier lugar modifica la forma en que el usuario interactúa con él. La interfaz de usuario de un monitor de ordenador y de un monitor de ordenador diminuto. La interfaz de usuario de un dispositivo móvil es algo diferente a la de un ordenador de sobremesa o portátil; los canales de comunicación también son muy distintos; los dispositivos móviles disponen de 3G, redes de voz, mensajería, geolocalización y videoconferencia inalámbrica. Aunque las redes ofrecen capacidades de datos de banda ancha, estas capacidades

pueden variar significativamente en función de la intensidad de la señal y la disponibilidad de la conexión de red, especialmente para los usuarios móviles.

### **Aplicación Nativa**

Las aplicaciones nativas están diseñadas exclusivamente para una plataforma, como el iPhone o Android, y pueden acceder a toda la funcionalidad del dispositivo, incluyendo la cámara, el GPS, la lista de contactos y las alertas del sistema. Se denominan aplicaciones nativas porque son "nativas" del dispositivo para el que fueron diseñadas. Son las aplicaciones que suelen estar disponibles a través de Google Play o la App Store de Apple. Son las más rápidas de los tres tipos de aplicaciones.

"Una aplicación nativa es aquella que se instala en el dispositivo y fue construida específicamente para su uso con un determinado sistema operativo". (Molina, Sandoval, & Toledo, 2012).

### **Aplicación Web**

Una aplicación web es un sitio web que realiza las mismas funciones que un programa nativo, pero no se instala o despliega de la misma manera que una aplicación nativa. Hacen uso de un navegador web y a menudo de HTML5. Además, dependen de un navegador web ampliamente utilizado para comprender el programa. Por otro lado, la frontera entre las aplicaciones nativas y las aplicaciones web se está difuminando a medida que más sitios web utilizan HTML5. Una aplicación web puede ofrecer varias ventajas sobre un programa nativo en términos de facilidad de desarrollo, portabilidad y compatibilidad entre plataformas.

Según Lujan Mora (2002) las aplicaciones Web son aquellas que permiten a los usuarios conectarse a un servidor Web a través de una red utilizando un determinado navegador. En consecuencia, se describe como una aplicación Web que es accesible a través de una red, ya sea intranet o Internet, a través de la Web. En general, las aplicaciones web se refieren a programas informáticos que se ejecutan a través del navegador.

## **Aplicación Híbrida**

El método híbrido mezcla el desarrollo nativo y el web. Esta estrategia permite a los desarrolladores crear la mayor parte de sus aplicaciones en tecnologías web para lograr la compatibilidad entre plataformas, pero manteniendo el acceso directo a los programas nativos cuando sea necesario. El componente nativo del programa hace uso de las API del sistema operativo para desarrollar un motor de búsqueda HTML incrustado que actúa como puente entre el navegador y las API del dispositivo. Este puente permite a la aplicación híbrida utilizar todas las capacidades disponibles en los smartphones actuales. Los desarrolladores de aplicaciones tienen la opción de desarrollar su propio puente o utilizar soluciones preconstruidas como PhoneGap, un paquete de código abierto que ofrece una interfaz JavaScript coherente para las funciones del dispositivo elegidas que son consistentes en todas las plataformas operativas. Aunque el componente nativo de la aplicación puede crearse individualmente, varias soluciones del mercado incluyen un contenedor nativo como parte de su oferta, lo que permite a los desarrolladores diseñar aplicaciones complejas que utilizan características completas del dispositivo utilizando sólo lenguajes web. En determinadas circunstancias, una solución permitirá al desarrollador utilizar los conocimientos nativos que pueda tener para adaptar el contenedor nativo a los requisitos específicos de la organización. La parte web del programa puede adoptar la forma de una página web del lado del servidor o una colección de archivos HTML, JavaScript, CSS y multimedia integrados en el código de la aplicación y mantenidos localmente en el dispositivo. Ambas estrategias ofrecen una serie de ventajas y desventajas. El código HTML almacenado en un servidor permite a los desarrolladores realizar pequeños cambios en una aplicación sin pasar por el procedimiento de presentación y aprobación que exigen algunas tiendas de aplicaciones. A continuación, se exponen las ventajas e inconvenientes del desarrollo de aplicaciones móviles híbridas (Puetate e Ibarra 2020).

## **Ventajas**

- a. Consumo de recursos del dispositivo y del sistema operativo.
- b. Los costes de desarrollo pueden ser menores que los de una especie nativa.
- c. Son independientes de la plataforma.
- d. Permiten la distribución a través de los escaparates de su plataforma.

## **Desventajas**

- a. La documentación puede ser limitada y desorganizada.

Una aplicación híbrida es exactamente lo que su nombre indica: una mezcla de una aplicación web y una aplicación nativa. "Residen" en una tienda de aplicaciones y pueden acceder a las funcionalidades del dispositivo de forma similar a la de un programa nativo.

Las aplicaciones híbridas intentan combinar las mejores características de las aplicaciones nativas y las aplicaciones web en una sola aplicación móvil. En pocas palabras, la aplicación se desarrolla como una Web App (utilizando HTML5, CSS y JS), y el framework se encarga de la conversión de la vista web a una vista de dispositivo móvil, convirtiendo todas las tareas de la vista web en actividades específicas del teléfono/tableta.

"De este modo, una Hybrid App puede llegar a numerosas plataformas móviles con poco esfuerzo, pero con el riesgo de perder el acceso a algunas de las características de cada dispositivo". (Angulo, R. 2013).

## **Lenguaje de programación**

Lenguaje formal (o artificial, es decir, con reglas gramaticales bien definidas) que permite a una persona, en este caso el programador, escribir (o programar) una serie de instrucciones o secuencias de comandos en forma de algoritmos para controlar el comportamiento físico o lógico de un sistema informático con el fin de obtener diversos tipos de datos o realizar determinadas tareas. Un programa de ordenador es una secuencia de comandos expresados en un lenguaje de programación (Joyanes, 2008).

## **Java**

Sun Microsystems comercializó inicialmente Java como lenguaje de programación y plataforma informática en 1995. Es casi seguro que numerosos programas y sitios web no funcionarían correctamente si no se instala Java, y cada día se desarrollan otros nuevos. Java es un lenguaje de programación rápido, seguro y fiable. Java es omnipresente, ya que funciona en todo, desde ordenadores portátiles hasta centros de datos, desde consolas de juegos hasta potentes ordenadores, desde teléfonos móviles hasta Internet; si funciona en una plataforma, no es necesario recompilarlo para que funcione en otra. En 2012, Java era uno de los lenguajes de programación más utilizados, sobre todo para aplicaciones cliente-servidor.

Según Javier Cruz (2012), Java, tal y como se entiende ahora, es un lenguaje de programación y un entorno para ejecutar aplicaciones Java. A diferencia de los compiladores convencionales, que convierten el código fuente en instrucciones a nivel de máquina, el compilador de Java convierte el código fuente de Java en instrucciones interpretables por la máquina virtual de Java (JVM). En comparación con C y C++, se inspira en. Java es un lenguaje de programación interpretado.

## **JavaScript**

En contra de la creencia popular, no hay que confundir Java con JavaScript. Java es un lenguaje de programación completo que permite el desarrollo de aplicaciones autónomas, mientras que JavaScript es un lenguaje de programación interpretado que se utiliza principalmente para el desarrollo de páginas web, ya que es similar tanto a Java como a C++, el lenguaje de programación utilizado por sistemas operativos como Windows. JavaScript es un lenguaje de scripting diseñado para integrarse con HTML y facilitar la construcción de sitios web interactivos sin necesidad de utilizar scripts avanzados o Java. Es un lenguaje de programación orientado a objetos que se creó con el fin de desarrollar aplicaciones cliente-servidor a través de Internet (Orós, 2002).

JavaScript puede insertarse en cualquier página HTML o en cualquier documento que se convierta en HTML en el navegador del cliente; esto incluye los documentos PHP y ASP. La incorporación directa de código en una estructura HTML es un método intrusivo que no se recomienda. La forma recomendada por el W3C es añadir JavaScript como un archivo externo, tanto por accesibilidad como por practicidad y velocidad de navegación.

### **Hyper Text Markup Language (HTML)**

HTML es un lenguaje de marcado que se estableció para organizar el material y especificar cómo debe mostrarse en formato de hipertexto, el estándar de la industria para los sitios de Internet.

Es un formato autónomo derivado de las etiquetas presentes en el Lenguaje de Marcas Generalizado Estándar (SGML). Este concepto, también denominado "Lenguaje de Marcado Generalizado Estándar", está diseñado para servir de marco de referencia para categorizar e identificar los numerosos documentos incluidos dentro de una lista. Su lenguaje se utiliza para proporcionar los nombres de las etiquetas que se utilizarán a lo largo del proceso de clasificación; como esta disposición no se rige por ninguna norma, se denomina sistema de formato abierto. (Pérez y Gardey, 2008).

#### **1.1.3. Normativa técnica**

##### **Normas ISO 25000**

Conocido como SQuaRE (Requisitos y Evaluación de Calidad de Sistemas y Software), ISO / IEC 25000 es una colección de normas destinadas a establecer un marco unificado para evaluar la calidad de los productos de software. La familia de normas ISO/IEC 25000 incorpora muchas normas anteriores, sobre todo la ISO/IEC 9126, que definen las características del modelo de calidad de los productos de software y el producto de software que se debe manejar. Esta familia de normas ISO/IEC 25000 se divide en cinco secciones.

#### A). ISO/IEC 2500n – División de Gestión de Calidad

Esta sección define todos los modelos genéricos, conceptos y definiciones a los que hacen referencia las demás normas de la familia 25000. Por el momento, esta división consta de los siguientes componentes:

**ISO/IEC 25000 - Guide to SQuaRE:** Incluye a. SQuaRE, familia de términos, descripción general de piezas, público objetivo y partes relacionadas, y modelos de referencia.

**ISO/IEC 25001 - Planning and Management:** Crear especificaciones y procedimientos para evaluar y gestionar las necesidades de productos de software.

#### B). ISO/IEC 2501n – División de Modelo de Calidad

Los estándares de esta sección proporcionan un modelo de calidad detallado, incluidas las características de calidad internas, externas y en uso del producto de software. Actualmente, esta división consta de los siguientes elementos:

**ISO/IEC 25010 - System and software quality models:** Se describen los modelos de calidad del software y los productos de calidad en uso. Esta norma internacional especifica las características y sub características de calidad que deben tenerse en cuenta al evaluar los productos de software.

**ISO/IEC 25012 - Data Quality model:** especifica un modelo general de calidad para los datos estructurados que es aplicable a los datos que se almacenan como parte de un sistema de información.

#### C). ISO/IEC 2502n – División de Medición de Calidad

Estas normas ofrecen modelos de referencia para evaluar la calidad de los productos, definiciones de las medidas de calidad (interna, externa y en uso) y recomendaciones para su uso. En la actualidad, este departamento se compone de lo siguiente:

**ISO/IEC 25020 - Measurement reference model and guide:** proporciona una descripción general de los factores de medición de calidad y un modelo

de referencia general. También proporciona orientación para que los usuarios seleccionen, desarrollen y apliquen los medios proporcionados por la norma ISO.

**ISO/IEC 25021 - Quality measure elements:** Establece y define una colección de líneas de base preferidas y métricas derivadas para su uso en todo el ciclo de vida del desarrollo de software.

**ISO/IEC 25022 - Measurement of quality in use:** establece una métrica para determinar la calidad del producto en el uso.

**ISO/IEC 25023 - Measurement of system and software product quality:** Se definen medidas específicas para evaluar la calidad de los productos y sistemas de software.

**ISO/IEC 25024 - Measurement of data quality:** define una métrica específica para medir la calidad de los datos.

#### **D). ISO/IEC 2503n – División de Requisitos de Calidad**

Los estándares que componen esta parte ayudan a definir los requisitos de calidad que se pueden utilizar en el proceso de lograr los requisitos de calidad para los productos de software en desarrollo o como una contribución al proceso de evaluación. Para lograr esto, esta sección incluye:

**ISO/IEC 25030 - Quality requirements:** ofrece una serie de sugerencias para determinar las normas de calidad de los productos de software.

#### **E). ISO/IEC 2504n – División de Evaluación de Calidad**

Esta sección abarca las normas que definen los requisitos, las sugerencias y las indicaciones para aplicar el proceso de evaluación de productos de software. Esta sección contiene lo siguiente:

**ISO/IEC 25040 - Evaluation reference model and guide:** presenta una metodología de evaluación estandarizada que tiene en cuenta las entradas del proceso de auditoría, las restricciones y los recursos necesarios para alcanzar cada conclusión.

**ISO/IEC 25041 - Evaluation guide for developers, acquirers and independent evaluators:** Requisitos y sugerencias para una evaluación satisfactoria de los productos de software desde la perspectiva de los desarrolladores, los compradores y los revisores independientes  
Descripción.

**ISO/IEC 25042 - Evaluation modules:** Definir lo que la norma entiende por módulos de evaluación y los documentos, estructuras y contenidos que deben utilizarse para definirlos.

**ISO/IEC 25045 - Evaluation module for recoverability:** define un módulo que evalúa las sub características de la resiliencia.

SQuaRE (de ISO / IEC 25050 a ISO / IEC 25099) está dedicada a estándares o informes técnicos relevantes para un área de aplicación en particular, o se puede utilizar además de otros estándares de la familia SQuaRE.

#### **1.1.4. Gestión de Riesgos**

Para la presente investigación se consideraron algunos riesgos sobre cuando tomar uno u otro framework para su debido análisis comparativo. Cabe mencionar que, si bien seleccionamos las muestras de manera no aleatoria, elegir cada uno de los frameworks a analizar supone un riesgo para el desarrollo de la investigación y su gestión está definida en este apartado.

Para la gestión de riesgos se toman los siguientes puntos:

**Prevención:** Es importante tener dentro de la gestión de riesgos la prevención, debido a que sería el primer paso para poder gestionar el riesgo de manera óptima, identificando la causa para de esta manera eliminarla y prevenirla para una posible reincidencia. La prevención es el paso clave para poder tener el control sobre alguna posible amenaza al desarrollo de la presente investigación.

**Mitigación:** Se debe tener claramente identificado que para poder mitigar el riesgo será necesario reducir al máximo las probabilidades de que este hecho ocurra, determinar con qué frecuencia podría ocurrir alguna amenaza

y qué casos particulares podrían darse, de manera que se pueda mitigar alguna causante o posible amenaza para evitar que el desarrollo de la investigación se vea perjudicado.

**Aceptación:** Para el peor de los casos, siendo que la amenaza se dé de manera inevitable, será necesario aceptar las consecuencias del riesgo desarrollando un plan de contingencia, así de esta manera se puede tomar un plan de acción a seguir en caso el riesgo llegue a ocurrir.

Lo que finalmente se quiere en este punto es indicar que se tiene de alguna manera el control de cómo lidiar con los riesgos que se presenten y que puedan perjudicar de una u otra manera el desarrollo óptimo del trabajo de investigación, así que se pueda garantizar que se podrán atender con eficacia y eficiencia la mayoría de problemas o riesgos que se presenten.

#### **1.1.5. Modelo de evaluación**

Como parte del modelo de evaluación para seleccionar las muestras dentro de la presente investigación se tomaron como referencia lo que dicen (Guncay Barzallo & Samaniego Mosquera, 2015) "Los modelos de evaluación son normas que han evolucionado a partir de la necesidad de cuantificar el grado de calidad y madurez del programa. Si bien el modelo de McCall fue el primero en publicarse en 1977, han surgido otros como el de Boehm, Gilb, FURPS, Dromey y la norma ISO 9126 para evaluar mediante diversas metodologías y por medio de diversos criterios los productos de programa. Además, los autores indican que, paralelamente a los modelos clásicos mencionados anteriormente, los modelos de evaluación del software de código abierto han surgido como una alternativa viable. Estos modelos son capaces de evaluar propiedades únicas y relevantes del software de código abierto que los modelos clásicos no evalúan, como la asociación, la adopción y las licencias. Hay una serie de técnicas ampliamente establecidas y estandarizadas para evaluar la calidad y la madurez de los productos de software de código abierto, muchas de las cuales se detallan aquí para ayudarle a elegir la más apropiada para la evaluación del marco.

### **1.1.6. Estado del Arte**

Bravo (2021) en su investigación “ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS, PARA EL DESARROLLO DE APLICACIONES MÓVILES MULTIPLATAFORMAS” indica que, en la década moderna, el mercado de dispositivos móviles inteligentes ha alcanzado su punto máximo, y este tipo de sistema de ingeniería se comunica gradualmente entre los usuarios y sus máquinas diarias, así como entre los usuarios. Con el continuo crecimiento de los dispositivos móviles, su utilidad se ha extendido a diferentes áreas, que incluyen no solo el uso de servicios web sino también la navegación por Internet, la creación de aplicaciones más actualizadas, etc.

El estudio surge a raíz de la problemática al momento del desarrollo de una aplicación móvil y su complejidad. El desarrollo de aplicaciones móviles puede ser difícil y costoso de acceder u obtener para las organizaciones que desean el plus en el momento de la compra. Una aplicación incluye todo lo que se necesita. Existen innumerables plataformas móviles disponibles, pero las más reconocidas son Android y iOS, siendo la misma, que se caracteriza por satisfacer algunas necesidades de cada cliente.

Esto no significa, sin embargo, que sean exclusivamente personales. Incluso hoy en día la gente todavía habla de HarmonyOs, una llamada para reemplazar el sistema operativo Android en el teléfono desarrollado por Huawei, y de hecho han pasado algunos años. Antes de que Android e iOS dominen el mercado móvil, había varias opciones para los sistemas operativos móviles.

El problema surge al momento de crear una aplicación móvil nativa, una desarrollada para una plataforma o equipo específico. No está disponible para un gran número de usuarios que utilizan otros sistemas operativos móviles. Si la misma aplicación está escrita para diferentes plataformas, ésta debe diseñarse en un idioma diferente, tecnologías diferentes, etc. En el caso de que se plantee crear la misma aplicación para diferentes plataformas, se deberá implementar utilizando otro lenguaje, entonces esto genera un gasto excesivo a la empresa, ya que surge la necesidad de

obtener un equipo técnico que se encuentre en la capacidad de poder analizar y comprender el lenguaje de programación, es decir, Java/ Kotlin para Android, y Swift / Objective-C para iOS.

En base a esto, puede ser conveniente evaluar los frameworks para el desarrollo de aplicaciones móviles multiplataforma. Esto permite anticipar el uso de estas herramientas para resolver las molestias discutidas anteriormente.

Como parte de los resultados de la investigación el autor dice que ambos frameworks comparados realizan un análisis activo de la legislación aplicada en el caso de estudio y solo satisfacen las necesidades de la aplicación para elegir qué tecnología o framework utilizar para el desarrollo de aplicaciones móviles multiplataforma. Considerando el tipo de aplicación que está desarrollando, la plataforma en la que se ejecuta la aplicación, etc. Ambos frameworks son excelentes opciones de desarrollo y están respaldados por grandes empresas como Facebook y Google. Por el contrario, está React Native, pero es un poco más maduro debido a su longevidad. Marlet.

Además, React Native se desarrolla utilizando el lenguaje actual como JavaScript, al mismo tiempo, las funciones Flutter y Dart son de alta calidad. Esto es especialmente cierto si se desea que la aplicación brinde la experiencia más original posible.

#### **1.1.7. Definición de términos**

**Android:** Google lo creó para dispositivos de pantalla táctil. Utilizando el núcleo de Linux como base. Desde 2013, es el sistema operativo móvil más vendido para tabletas y smartphones.

**Backend:** Conocido como en el lado del servidor, se denomina de esta manera a todo el código que corre dentro de un servidor y no es visible ante el usuario, aquí funciona la aplicación, servidor y base de datos.

**Css:** Las hojas de estilo en cascada son un lenguaje de programación que explica cómo se presentan los textos estructurados en hojas de estilo para varios tipos de interpretación.

**Consumo de recursos de PC:** Hace referencia al consumo de recursos de una PC (Consumo de memoria RAM y el porcentaje de uso del procesador en todos los núcleos) que demanda un proceso o servicio informático.

**CRUD:** En informática, CRUD es el acrónimo de "Crear, Leer, Actualizar y Eliminar" se refiere a las operaciones fundamentales de las bases de datos o la capa de persistencia en el software.

**Curva de aprendizaje:** Se refiere al grado de logro alcanzado a lo largo del tiempo de estudio. Es un gráfico en el que el eje horizontal indica el tiempo y el eje vertical refleja el número de éxitos durante ese periodo de tiempo. A menudo se cometen numerosos errores al iniciar una nueva actividad.

**Flujo:** Procedimiento que sigue un programa para la resolución de un problema.

**Frontend:** Es la parte de la aplicación que es visible ante los usuarios, conocida como en el lado del cliente. Es todo lo que se ve en pantalla, desde colores, tipos de letra, imágenes, efectos del mouse, desplazamientos y otros efectos visuales que permitan navegar por una página web.

**H2 database:** Es un administrador de base de datos relacional programado en java.

**Horas de esfuerzo:** una estimación aproximada de la cantidad de esfuerzo humano que un trabajador típico puede invertir en una hora en una actividad, proceso o proyecto.

**HTTP:** Hypertext Transfer Protocol o Protocolo de Transferencia de Hipertexto en su traducción al español, es un protocolo a nivel de aplicación que permite la transmisión de documentos HTML.

**Implementación:** Se refiere al proceso de instalación, implementación o ejecución de un programa informático.

**iOS:** Es un sistema operativo móvil desarrollado por Apple, originalmente creado para iPhone, iPad y el iPod touch. Apple no permite que este sistema operativo sea instalado en hardware de terceros.

**Jmeter:** Es un software que puede ser usado como herramienta para realizar pruebas de sobrecarga y estrés a las aplicaciones web.

**Modelo:** Es la representación gráfica de los datos sobre los que funciona el sistema.

**Microservicios:** Son un enfoque como parte de la arquitectura en el desarrollo de software donde cada servicio actúa de manera independiente. Cada microservicio puede tener su propia base de datos.

**Megabytes:** Medida de la capacidad de memoria de una computadora que es igual a 1 millón de bytes.

**Milisegundo:** Medida de tiempo que es igual a la milésima parte de un segundo.

**Proceso de instalación:** Se define al proceso de instalación de un programa o software en una PC.

**Springboot:** Según su página oficial, se puede definir como una solución para crear aplicaciones basadas en Spring de una manera rápida, autónoma y con características deseables para producción.

**Servicio web:** Es una tecnología que utiliza protocolos y estándares para intercambiar información entre aplicaciones de manera independiente al lenguaje con el que hayan sido creadas las aplicaciones.

**Tiempo de respuesta:** Es el tiempo que le lleva a un sistema o la unidad funcional reaccionar a una entrada dada.

**Tomcat:** Funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta, funciona como servidor de aplicaciones.

**Vista:** Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

#### **1.4. Formulación del Problema.**

¿Cómo elegir el framework que mejor se adapte a las necesidades de un equipo

de desarrollo a la hora de implementar una aplicación móvil híbrida?

### **1.5. Justificación e importancia del estudio.**

El desarrollo de software está presente en el ámbito académico ya que en la actualidad se desarrollan diversas aplicaciones para algunas instituciones educativas, como colegios primarios, secundarios y universidades o institutos, el uso de un adecuado Framework para el desarrollo de las aplicaciones móviles híbridas nos dará una cierta ventaja al momento de poder ejecutar un proyecto para éste ámbito, conocer y explotar esta herramienta es de suma importancia para los profesionales de la tecnología, ya que nos permitirá reducir costos y maximizar ganancias en cuanto a tiempo de desarrollo.

El análisis comparativo de los frameworks de desarrollo para aplicaciones móviles híbridas siempre será de un gran avance para la tecnología ya que podremos medir estas herramientas para tener en cuenta, evaluando sus ventajas y desventajas cuál de ellos es el que mejor se adapta a nuestro desarrollo, el que mejor cumple con nuestras expectativas y se ajusta a nuestro proyecto.

Al conocer los frameworks y estar convencido de cuál es el más adecuado para nuestro desarrollo, aporta al ámbito social ya que se podrá desarrollar manuales de usuario, utilitarios, etc. Además, se dará a conocer los resultados obtenidos para próximas investigaciones, así como también se aporta de manera científica al tener una idea más clara en cuanto a medición y eficiencia de cada uno de los frameworks para su posterior uso.

En el ámbito económico repercute de manera importante ya que en las empresas de desarrollo tecnológico, así como otras entidades que deseen implementar alguna aplicación haciendo uso de estas herramientas tecnológicas podrán tener una reducción de costos y maximizar ganancias, ya que se estará explotando la herramienta de manera óptima y el desarrollo del proyecto será satisfactorio para ambas partes tanto como para los desarrolladores como para el cliente.

Este análisis comparativo entre los frameworks Ionic y Apache Cordova para el desarrollo de aplicaciones móviles híbridas ayudará a poder comprender el por

qué cada vez más las aplicaciones móviles híbridas están desplazando a las aplicaciones desarrolladas para plataformas específicas.

Angulo, R. (2013) afirma: “Una aplicación híbrida se basa en el desarrollo de una página móvil con capacidad para manejar los elementos nativos del dispositivo (cámara y GPS, entre otros).”

## **1.6. Hipótesis.**

Al analizar y comparar ambos marcos de desarrollo para aplicaciones móviles, a través de la creación de aplicaciones con funcionalidad CRUD en cada marco y el establecimiento de casos de prueba, se espera que ninguno sea superior, ya que cada uno tiene sus propias características, así como su propio conjunto de ventajas y desventajas.

Mientras que Apache Cordova es más agnóstico en cuanto a plataformas que Ionic, este último trabaja con el Framework Angular, que es un producto de Google con amplia documentación, conocimiento y soporte. Apache Cordova puede utilizarse junto con cualquier otro framework JS o CSS.

Ambos son de código abiertos, lo que significa que pueden ser difundidos libremente, lo cual es una ventaja importante para cualquier marco que se elija para desarrollar una aplicación móvil.

Ionic tiene una ventaja en términos de simplicidad de uso, configuración y administración; también vale la pena señalar que ambos marcos se ejecutan en Apache Cordova.

## **1.7. Objetivos.**

### **1.1.8. Objetivo General**

Realizar análisis comparativo de los frameworks para el desarrollo de aplicaciones móviles híbridas.

### **1.1.9. Objetivos específicos**

- a. Seleccionar los frameworks para realizar la comparativa.
- b. Desarrollar una aplicación como caso de estudio para la comparación.
- c. Realizar pruebas a las aplicaciones desarrolladas con base en los frameworks.

## **II. MATERIAL Y MÉTODOS**

### **2.1. Tipo y diseño de investigación.**

Es una investigación Aplicada con su diseño Cuasi Experimental, la misma que se desarrolló bajo el enfoque Cuantitativo, según José Zelada (2014) manifiesta que la investigación aplicada tiene como objetivo generar conocimiento que se pueda aplicar directamente a problemas en los campos de la sociedad y la producción. Se basa esencialmente en los descubrimientos técnicos de la investigación básica que se ocupa del proceso de conexión entre teoría y productos.

### **2.2. Población, muestra y muestreo**

#### **2.2.1. Población**

Para la presente investigación se tomó como población los mejores frameworks de desarrollo para aplicaciones móviles híbridas vigentes actualmente en el mercado.

#### **2.2.2. Muestra**

Al ser una investigación de diseño Cuasi Experimental se ha elegido la muestra no aleatoria, se presentan las siguientes:

- Ionic Framework
- Apache Cordova

Para la selección de la muestra se tomaron en cuenta criterios como software free, fácil acceso a documentación y material videográfico.

### **2.3. Variables y Operacionalización**

Considerando las variables definidas para esta investigación y en base a los objetivos propuestos se presenta el siguiente cuadro de operacionalización de variables.

Tabla 1.  
Matriz de operacionalización de variables

Variable	Dimensión	Indicadores	Índice	Técnica de recolección	Instrumento de recolección
Framework	Frameworks híbridos	Frameworks híbridos vigentes	Cantidad de Frameworks	Análisis documental	Guía de análisis documental
	Curva de aprendizaje	Número de horas/esfuerzo para desarrollo del módulo.	Horas de esfuerzo	Entrevista	Cuestionario
	Proceso de instalación	Nivel de dificultad en la instalación	Bajo Intermedio Alto	Entrevista	Cuestionario
		Consumo de memoria RAM (Memoria física usada por los procesos activos).	Megabytes (MB)	Observación	Guía de observación.
	Consumo de recursos de PC	Consumo de % CPU (Uso del procesador en todos los núcleos)	Porcentaje de uso del CPU	Observación	Guía de observación.
Aplicaciones móviles híbridas	Consumo de recursos de PC	Consumo de memoria RAM (Memoria física usada por los procesos activos). Consumo de % CPU (Uso del procesador en todos los núcleos)	Megabytes (MB)	Observación	Guía de observación.
		Consumo de % CPU (Uso del procesador en todos los núcleos)	Porcentaje de uso del CPU	Observación	Guía de observación.
	Velocidad de respuesta	Tiempo de respuesta	Milisegundos		

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

### **Técnicas de recolección de datos**

Para la presente investigación se utilizaron las siguientes técnicas:

**Entrevista;** Este enfoque se utiliza para recoger datos de un solo individuo a la vez. Permite una comunicación más fluida y la obtención de información más detallada. Utiliza como herramienta un cuestionario o una guía de entrevista.

**Observación;** tiene el objetivo de recopilar datos sobre acontecimientos, hechos y procesos, entre otros. Se suele emplear cuando no sólo es fácil indagar sobre detalles concretos, sino también corroborarlos directamente. La guía de observación le sirve de herramienta.

**Análisis documental;** técnica de recopilación de datos donde se hace uso de herramientas manuales o automatizadas y se extrae la información de bases de datos, archivadores y demás fuentes de información valiosa para la investigación.

Según Roberto Hernández Sampieri, Carlos Fernández y María del Pilar Baptista, "La investigación documental es el proceso de identificación, obtención y consulta de biografías y otros materiales que provienen de otros conocimientos y/o información que se recoge selectivamente de cualquier realidad para que sea relevante para los objetivos del estudio."

### **Instrumentos de recolección de datos**

En este estudio se utilizaron los siguientes instrumentos:

**Cuestionario;** es un documento compuesto por una serie de preguntas donde se crean, plantean, ordenan y estructuran constantemente según un plan particular. Esto le dará toda la información que necesita en su respuesta.

**Guía de observación;** Es una herramienta que permite al observador situarse sistemáticamente en lo que realmente se investiga. También es un medio para recopilar datos e información sobre eventos y fenómenos.

**Guía de análisis documental;** este instrumento permite que se pueda recopilar información de diferentes fuentes de datos mediante la documentación, se analiza las fuentes de información como bases de datos, sitios web confiables, etc.

#### **2.4.1. Confiabilidad de instrumentos**

Según esta idea, la fiabilidad se refiere al grado en que el instrumento produce el mismo valor cuando se utiliza "n" veces.

Del mismo modo, los instrumentos elaborados para este estudio han sido sometidos a evaluaciones de fiabilidad mediante el Alfa de Cronbach.

El Alfa de Cronbach es una medida de fiabilidad de la escala que Cronbach acuñó en 1951.

El alfa de Cronbach es la media de las correlaciones entre las variables de la escala y puede determinarse de dos maneras: por la varianza (alfa de Cronbach) o por la correlación de los ítems (alfa de Cronbach normalizado). El factor alfa puede utilizarse para indicar la fuerza interna, pero no transmite ninguna información sobre la estabilidad temporal del instrumento o la equivalencia entre las diferentes formas.

#### **Reportes de Análisis de Confiabilidad mediante el Alfa de Cronbach**

**Cuestionario 01:** Cuestionario para determinar la curva de aprendizaje.

Mediante este cuestionario se pretende medir la experiencia del desarrollador utilizando cada uno de los frameworks seleccionados como muestra para la investigación, considerando la curva de aprendizaje, las horas de esfuerzo utilizadas y todo el proceso que conlleva familiarizarse con los frameworks en el proceso de desarrollo de las apps

El análisis de Alfa de Cronbach para validar este instrumento arrojó información que se presenta a continuación en las siguientes tablas.

Tabla 2.  
*Estadísticas de escala para cuestionario 01*

<b>Media</b>	<b>Varianza</b>	<b>Desviación estándar</b>	<b>N° de elementos</b>
15.00	12.000	3.464	9

*Nota:* La información corresponde a los datos arrojados mediante el análisis del Alfa Cronbach utilizando el software SPSS. Fuente: Elaboración propia.

Tabla 3.  
*Estadísticas de fiabilidad para cuestionario 01*

<b>Alfa de Cronbach</b>	<b>N° de elementos</b>
0.906	9

*Nota:* La información corresponde a los datos arrojados mediante el análisis del Alfa Cronbach utilizando el software SPSS. Fuente: Elaboración propia.

**Cuestionario 02:** Cuestionario para medir la dificultad de proceso de instalación

Con este cuestionario también aplica y considera la experiencia del desarrollador con el uso de los frameworks seleccionados, en este caso basados en todo lo que sería el proceso que conlleva la descarga e instalación.

Tabla 4.  
*Estadísticas de escala para cuestionario 02*

<b>Media</b>	<b>Varianza</b>	<b>Desviación estándar</b>	<b>N° de elementos</b>
9.75	2.250	1.500	7

*Nota:* La información corresponde a los datos arrojados mediante el análisis del Alfa Cronbach utilizando el software SPSS. Fuente: Elaboración propia.

Tabla 5.  
*Estadísticas de fiabilidad para cuestionario 02*

<b>Alfa de Cronbach</b>	<b>N° de elementos</b>
0.778	7

*Nota:* La información corresponde a los datos arrojados mediante el análisis del Alfa Cronbach utilizando el software SPSS. Fuente: Elaboración propia.

Bajo los resultados que devuelve el análisis de Alfa de Cronbach se puede evidenciar que los instrumentos utilizados en la presente investigación cumplen con la fiabilidad mínima requerida, de tal manera que se puede decir que los instrumentos son confiables.

#### **2.4.2. Validación de instrumentos**

Para determinar la validez de los instrumentos que se han utilizado en la investigación es necesario que los mismos cumplan con lo que se pretende medir, es decir que los instrumentos a validar puedan dar a conocer información relevante con respecto a las variables del estudio en desarrollo; se ha solicitado la validación de los instrumentos a expertos de modo que se puede asegurar que las preguntas en el cuestionario no contenga errores comunes, como preguntas confusas o ambiguas, con esta validación se espera que la investigación permita cumplir los objetivos propuestos así mismo que se cumplan los criterios éticos y de rigor científico.

#### **2.5. Procedimiento de análisis de datos**

En este caso el procedimiento usado fue analizar cada uno de los diferentes frameworks de desarrollo de aplicaciones móviles híbridas vigentes y más usadas en el mercado a fin de validar e identificar la población y muestra para la investigación, así mismo en base a las frameworks analizados se tomarán dos de ellos para instalar e implementar una aplicación móvil a fin de comparar y

medir cada uno de los frameworks en cuanto a curva de aprendizaje, horas de esfuerzo, dificultad de instalación así como los resultados obtenidos con la aplicación implementada midiendo consumo recursos de PC, tiempo de respuesta de la aplicación, etc., para todo lo anterior se usarán los instrumentos de cuestionario y guía de observación.

A continuación, se detalla cómo se aplicaron las técnicas y los instrumentos en la investigación, el siguiente cuadro muestra de forma descriptiva el procedimiento de análisis de los datos.

Tabla 6.  
*Procedimiento de análisis de datos*

<b>N°</b>	<b>Técnica</b>	<b>Instrumento</b>	<b>Fuente/ Informante</b>	<b>Descripción</b>
1	Entrevista	Cuestionario	Desarrolladores de aplicaciones	Entrevistar a desarrolladores de aplicaciones móviles para obtener información relevante sobre la experiencia que tuvieron en el desarrollo de las aplicaciones usando cada uno de los frameworks definidos como parte de la muestra de la investigación.
2	Observación	Guía de Observación	Frameworks/Aplicación implementada	Recopilar datos sobre el uso de determinados frameworks al diseñar aplicaciones móviles híbridas; esto nos permitirá comprender mejor el rendimiento de nuestras herramientas temáticas, lo que nos permitirá identificar sus

3	Análisis documental	Guía de Análisis documental	Artículos científicos	<p>ventajas y desventajas. Además, se identifican medidas para el uso de recursos y el tiempo de reacción.</p> <p>Revisión de artículos científicos donde se pueda validar los frameworks de aplicaciones móviles híbridas vigentes y más utilizados actualmente.</p>
---	---------------------	-----------------------------	-----------------------	---

A continuación, se muestra la matriz de objetivos específicos e instrumentos de recolección de información.

Tabla 7.

*Matriz de objetivos específicos e instrumentos de recolección de información*

<b>ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS DE DESARROLLO PARA APLICACIONES MÓVILES HÍBRIDAS</b>				
<b>Instrumentos de recolección de datos</b>	<b>OE-1</b>	<b>OE-2</b>	<b>OE-3</b>	
Guía de análisis documental para encontrar los diferentes frameworks híbridos móviles.	x			
Cuestionario para determinar la curva de aprendizaje		x	x	
Cuestionario para medir la dificultad de proceso de instalación		x	x	
Guía de observación para medir consumo de recursos de PC (Frameworks)				x
Guía de observación para medir consumo de recursos de PC (Aplicación)				x
Guía de observación para medir el tiempo de respuesta				x

**OE-1** Seleccionar los frameworks para realizar la comparativa.

**OE-2** Desarrollar una aplicación como caso de estudio para la comparación.

**OE-3** Realizar pruebas a las aplicaciones desarrolladas con base en los frameworks.

### 2.5.1. Métodos de análisis de datos

En base a la información cuantitativa, ésta fue procesada mediante hojas de cálculo, se procedió a ingresar la información al software de manera que este procesó los datos para mostrar la información a modo de cuadros y gráficos.

Para procesar la información se utilizaron las siguientes fórmulas:

**La Sumatoria;** Esta fórmula, también conocida como operación de suma, la notación sigma o el símbolo de la suma, nos permite calcular el resultado de la suma de muchos sumandos, e incluso de infinitos sumandos., se expresa con la letra griega en mayúscula sigma ( $\Sigma$ ).

$$\sum_{i=m}^n a_i = a_m + a_{m+1} + a_{m+2} + \dots + a_n$$

**La Media;** es la media de las muestras y es independiente de las amplitudes de los intervalos. Se denomina y sólo se encuentra en las variables cuantitativas. Se calcula juntando todos los valores y dividiendo por el número total de puntos de datos.

$$Media(X) = \bar{x} = \frac{X_1 + X_2 + \dots + X_N}{N}$$

### 2.6. Criterios éticos

Considerando como base lo que se indica en el Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica, en esta investigación tiene en cuenta los criterios éticos que se detallan a continuación.

### **2.6.1. Ambiental**

Con un amplio compromiso con el cuidado del medio ambiente, la investigación contempla el uso de material informático que no dañe o deteriore el medio ambiente mediante el uso de tecnología punta, así como el fomento del desarrollo de aplicaciones móviles que contribuyan al cuidado del medio ambiente al evitar el uso de papel.

### **2.6.2. Objetividad**

El escenario observado a lo largo de la investigación se presentará con un alto grado de objetividad, a la vez que se ofrece información para un estudio adicional y la mejora de la investigación.

### **2.6.3. Originalidad**

Cada pieza de material ofrecida en este estudio será referenciada, demostrando que no ha habido robo intelectual.

### **2.6.4. Veracidad**

La información aquí mostrada es totalmente verdadera cuidando la confidencialidad de la misma.

## **2.7. Criterios de Rigor Científico**

Este estudio se ajusta a los siguientes requisitos de rigor científico.

**Fiabilidad;** los datos obtenidos en la investigación son fiables, son datos que han sido validados por expertos a modo que se puede decir que la investigación cumple con este criterio. Así mismo si se replica la investigación utilizando los mismos métodos o estrategias los resultados obtenidos serán similares.

**Validez;** Los datos recogidos se verifican ya que, antes de la recogida de datos, se creó una proyección para determinar el tipo de resultados que se producirían, lo que garantiza que los datos se validen adecuadamente según las variables de Operacionalización en esta fase.

**Credibilidad;** denominado autenticidad, se refiere a la aproximación que los resultados de esta investigación tienen relación con el caso observado.

**Aplicabilidad;** la investigación puede ser llevada a otro contexto.

**Replicabilidad;** es posible repetir esta investigación.

### III.RESULTADOS

#### 4.1.Presentación de resultados

Considerando las variables de la investigación y los objetivos específicos se presentan los siguientes resultados:

##### 4.1.1. Seleccionar los frameworks para realizar la comparativa.

Para obtener los resultados de este objetivo Como parte del estudio, se empleó el enfoque de análisis documental junto con la guía de análisis documental, que incluía el análisis de varias fuentes bibliográficas que cumplieran el objetivo de la investigación. A continuación, se presenta el cuadro de resultados.

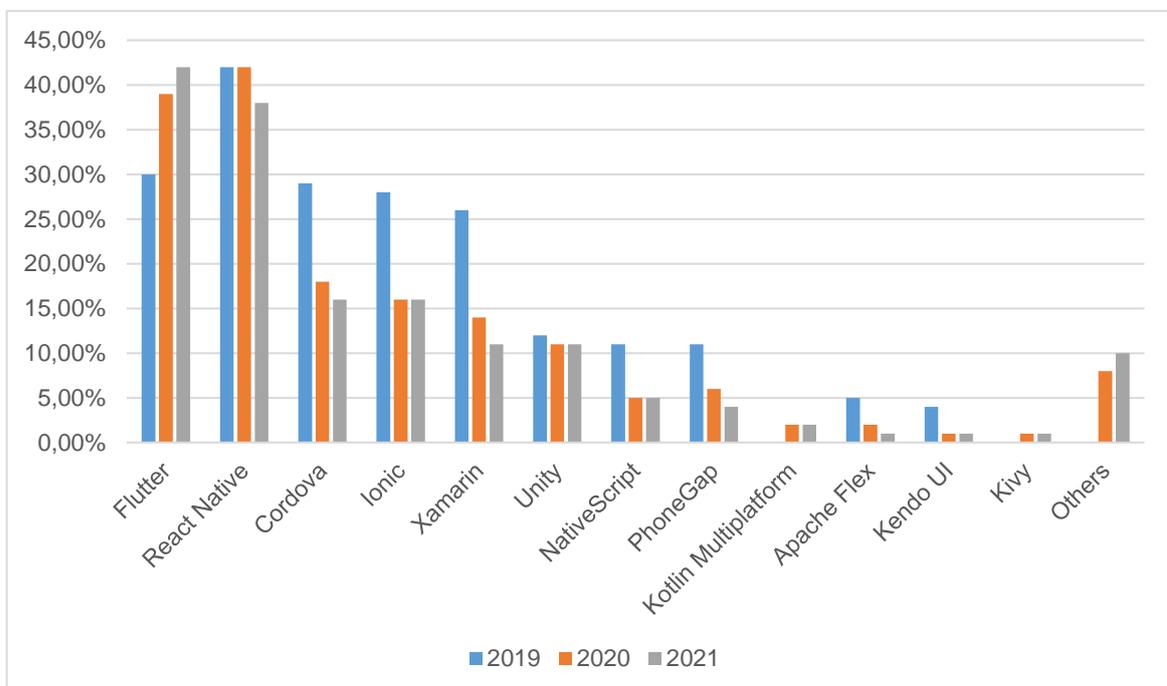


Figura 5. Frameworks móviles híbridos utilizados por desarrolladores de

software durante los años 2019 a 2021. Fuente: (Varun Bhagat, Hybrid Mobile App Frameworks You Can Opt for in 2021 & Beyond, 2021)

Según la gráfica se puede visualizar que entre los frameworks más utilizados durante los últimos tres años se encuentran Flutter, ReactNative, Apache Cordova y Ionic.

Según estos resultados obtenidos se tomó en cuenta a los framework **Apache Cordova y Ionic** como muestra para la presente investigación.

Los criterios que se consideraron para seleccionar la muestra son porque los frameworks contienen bastante documentación en sus sitios oficiales, así como material disponible (tutoriales, videos, ejemplos, etc.) en la red.

#### **4.1.2. Desarrollar una aplicación como caso de estudio para la comparación.**

Para obtener los resultados que cumplen con el segundo objetivo de la investigación presentada, se ha desarrollado una aplicación móvil con cada uno de los frameworks seleccionados como muestra, es importante señalar que como parte de recoger información con respecto a la experiencia del desarrollador se ha utilizado como técnica la encuesta haciendo uso del instrumento del cuestionario, así mismo para medir el consumo de recursos por cada framework se tiene como instrumento la guía de observación.

El desarrollo de un CRUD por cada una de las aplicaciones creadas con ambos frameworks seleccionados como parte de la muestra permite poder medir las dimensiones que corresponden detalladas anteriormente en la matriz de operacionalización de variables.

Se ha determinado un realizar un CRUD ya que es un procedimiento casi imprescindible en el desarrollo de cualquier aplicación web o proyecto de desarrollo de software, así mismo mediante un CRUD se puede transparentar el esfuerzo y la experiencia en la interacción con cada uno de los frameworks a comparar.

A continuación, se detallan los resultados obtenidos durante el proceso de desarrollo de las aplicaciones con los frameworks utilizados al momento de la creación del CRUD.

- a) Curva de aprendizaje:** Mediante encuestas realizadas a diferentes programadores, se pudieron obtener los resultados que se muestran a continuación con respecto a la curva de aprendizaje basado en horas de esfuerzo al implementar servicios web para la creación del CRUD correspondiente, así como las diferentes interfaces gráficas con el manejo de cada uno de los frameworks seleccionados para el análisis comparativo en la presente investigación.

Tabla 8.  
*Horas de esfuerzo por Framework para creación del CRUD*

<b>Framework</b>	<b>Create</b>	<b>Read</b>	<b>Update</b>	<b>Delete</b>	<b>Unidad de medida</b>
IONIC FRAMEWORK	16	16	12	8	Horas
APACHE CORDOVA	16	15	10	10	Horas

*Nota:* Los valores mostrados corresponden a las horas de esfuerzo para el desarrollo de cada una de las operaciones CRUD. Fuente: Elaboración propia.

Así mismo se pudo evidenciar que para ambos frameworks resulta fácil obtener documentación en la web, así como de fácil integración con el proyecto.

Tabla 9.  
*Dificultad para obtener documentación e integración al proyecto*

<b>Framework</b>	<b>Dificultad para obtener documentación (Fácil, Intermedio, Difícil, Muy difícil)</b>	<b>Dificultad para integración al proyecto (Fácil, Intermedio, Difícil, No hay material)</b>
IONIC FRAMEWORK	Fácil	Fácil
APACHE CORDOVA	Fácil	Fácil

*Nota:* Dificultad para obtener documentación en la web (Página oficial de cada

uno de los Frameworks). Fuente: Elaboración propia.

Tabla 10.

*Número total de horas de esfuerzo para desarrollar CRUD*

Framework	CRUD TOTAL	Unidad de medida
IONIC FRAMEWORK	52	Horas
APACHE CORDOVA	51	Horas

*Nota:* Sumatoria total de horas esfuerzo para la elaboración del CRUD completo por cada uno de los frameworks. Fuente: Elaboración propia.

Según la entrevista realizada a los programadores se puede evidenciar que la curva de aprendizaje medida en horas de esfuerzo para cada uno de los frameworks es similar con la ligera diferencia de una hora de Apache Cordova sobre Ionic.

Con esta pequeña diferencia se puede llegar a concluir que ambos frameworks son fáciles de aprender y la curva de aprendizaje es baja, puesto a que se puede lograr implementar servicios web y funcionalidades en corto tiempo.

Adicional a eso se puede evidenciar que no existe dificultad alguna para encontrar documentación referente a cada uno de los frameworks utilizados en la investigación en la web, en primera instancia en su página oficial, puesto a que los resultados muestran que es fácil encontrar documentación y material disponible para cada uno de ellos, de esta manera disminuyendo significativamente la curva de aprendizaje.

Finalmente, la documentación también incluye tutoriales para su fácil integración con los proyectos de desarrollo de aplicaciones móviles híbridas

**b) Nivel de dificultad de la instalación:** Para obtener estos resultados se realizaron encuestas a programadores sobre cómo fue su experiencia al momento de realizar la instalación de cada uno de los frameworks, de

manera que se muestran los resultados en base a la opinión de cada uno.

Tabla 11.  
*Nivel de dificultad de la instalación por Framework*

<b>Framework</b>	<b>Dificultad para encontrar instalador (Fácil, Intermedio, Difícil, Muy difícil)</b>	<b>Se puede instalar mediante consola CMD (sí, no)</b>
IONIC FRAMEWORK	Fácil	Sí
APACHE CORDOVA	Fácil	Sí

*Nota:* Se puede evidenciar el nivel de dificultad para la instalación por cada una los frameworks. Fuente: Elaboración propia.

Se puede concluir que no existe mayor dificultad para lograr encontrar el instalador adecuado para cada framework en la red y existe material disponible para su fácil instalación, se tiene como resultado que es fácil encontrar el instalador, así como su fácil descarga.

Adicional a lo anterior cada uno de los framework cuentan con la facilidad de poderse instalar/integrar con los proyectos frontend mediante consola o CMD, de esta manera facilita al desarrollador con el proceso de instalación e integración correspondiente.

- c) Consumo de recursos:** Mediante una guía de observación se pudo validar el consumo de recursos cuando se está ejecutando cada uno de los frameworks, se muestra la siguiente tabla:

Tabla 12.  
*Consumo de recursos de PC por Framework*

<b>Framework</b>	<b>Consumo de RAM (Megabytes)</b>	<b>% de CPU</b>
IONIC FRAMEWORK	68.50 MB	1.8%
APACHE CORDOVA	64.20 MB	1.6%

*Nota:* La computadora donde se realizaron las pruebas tiene un procesador con

las siguientes características: 11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80GHz  
2.80 GHz. Fuente: Elaboración propia.

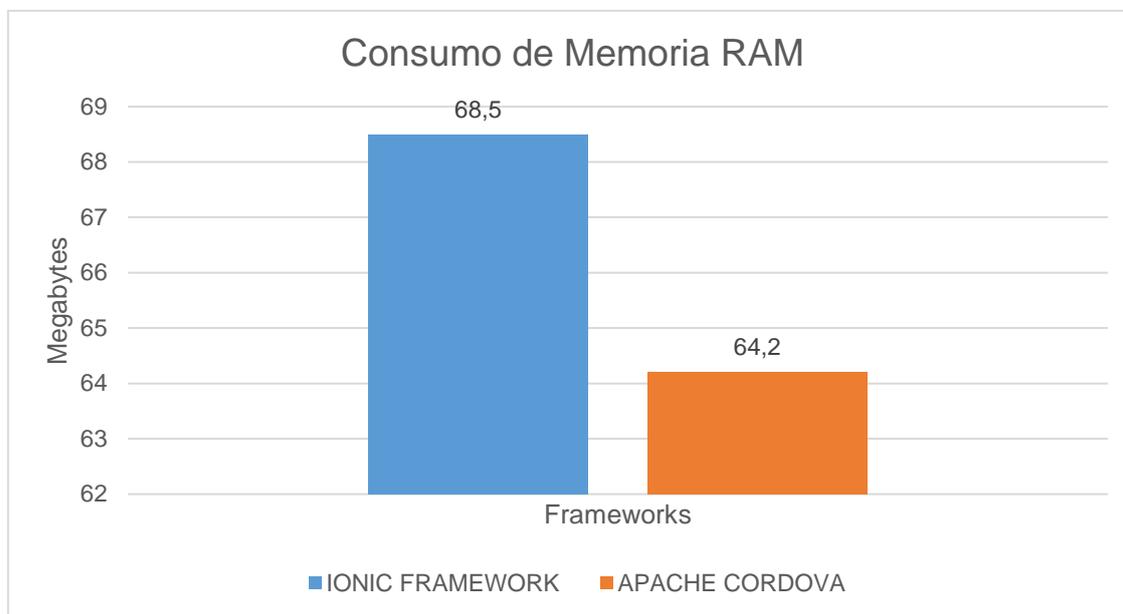


Figura 6. Consumo de memoria RAM por cada uno de los frameworks. Fuente: Elaboración propia.

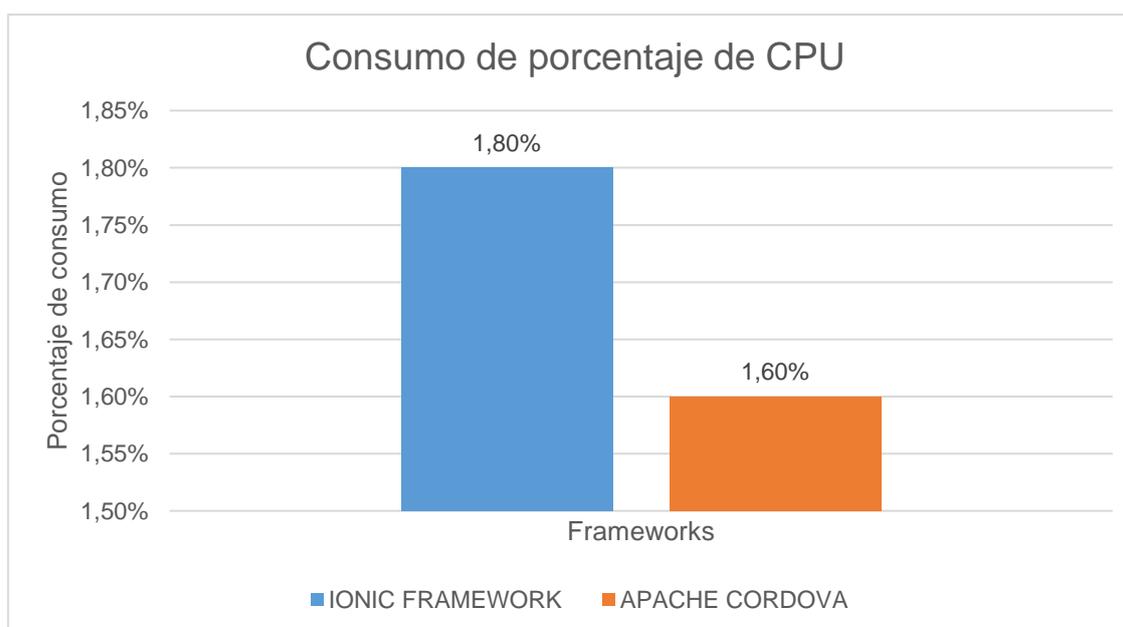


Figura 7. Consumo de porcentaje de CPU por Framework. Fuente: Elaboración Propia.

Como se puede visualizar en los resultados presentados en esta sección se puede observar que Apache Cordova es el framework que consume menos recursos tanto en memoria RAM, así como el porcentaje de consumo de CPU, pero con una diferencia mínima, ya que para los entornos de desarrollo es un número dato importante para poder elegir uno de los dos framework que se están analizando.

En este caso dependerá mucho de la capacidad de la PC donde se esté trabajando y la cantidad de programas o herramientas que se esté utilizando en simultáneo.

Finalmente, como conclusión de este espacio se tiene que los resultados obtenidos en esta sección muestran que Apache Cordova es más ligero por lo que consume menos recursos de la PC durante el entorno de desarrollo.

#### **4.1.3. Realizar pruebas a las aplicaciones desarrolladas con base en los frameworks.**

Para lograr obtener los resultados obtenidos para este objetivo, se ha desarrollado una aplicación con un CRUD que pueda gestionar usuarios, se destaca la creación de un CRUD para la gestión de usuarios ya que es un proceso de suma relevancia en los proyectos de desarrollo de software, casi todas las aplicaciones que se desarrollan intervienen procesos CRUD, de tal manera al momento de crear las aplicaciones con cada uno de los frameworks seleccionados se pueda medir el consumo de recursos que consume cada framework, para recoger esta información se utilizó la técnica de la observación mediante su instrumento de guía de observación, a continuación se detallan los resultados obtenidos.

Cabe mencionar que para obtener los resultados como parte de este objetivo fue necesario desarrollar un plan de pruebas con una concurrencia de 50 usuarios y se ha medido con la herramienta Jmeter.

Luego de ejecutar el plan de pruebas sobre las aplicaciones con CRUD

implementadas se muestran los resultados luego de aplicar la técnica de la observación, mediante su instrumento de la guía de observación. Se presenta a continuación algunos puntos de los que se han considerado para tomar métricas dentro de la investigación.

- a. **Consumo de recursos:** Para obtener estos resultados se ha utilizado el instrumento de la guía de observación a través del administrador de tareas de Windows y la herramienta Jmeter.

Tabla 13.

*Consumo de recursos de PC por aplicación desarrollada - Android*

<b>Framework</b>	<b>Consumo de RAM (Megabytes)</b>	<b>Porcentaje de CPU</b>
IONIC FRAMEWORK	50.45 MB	1.4%
APACHE CORDOVA	60.65 MB	1.8%

*Nota:* La computadora donde se realizaron las pruebas tiene un procesador con las siguientes características: 11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80GHz 2.80 GHz. Fuente: Elaboración propia.

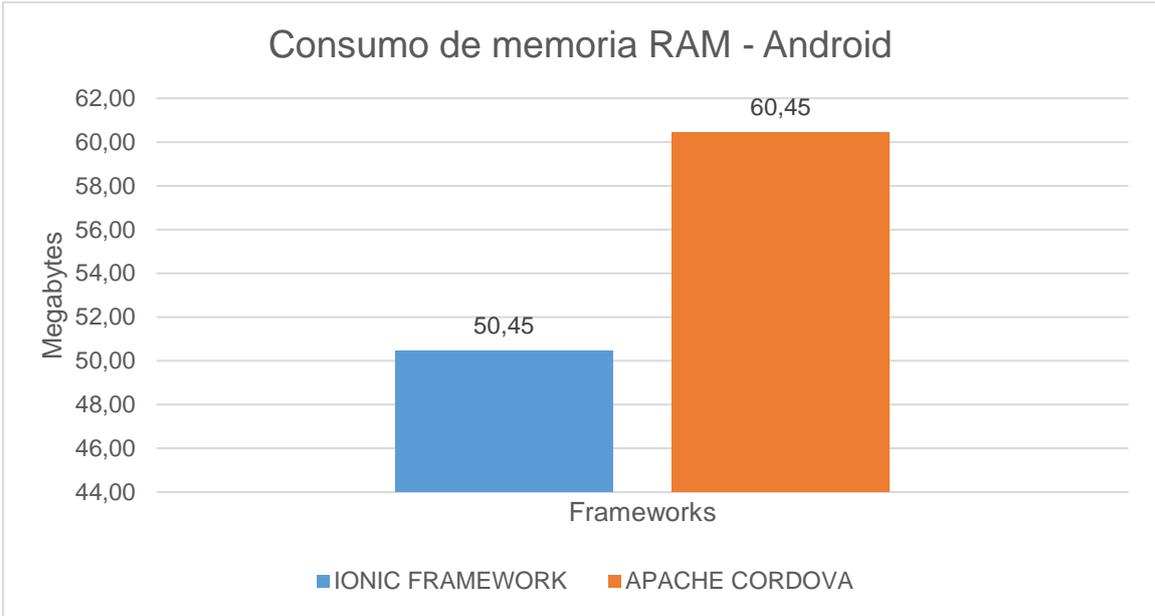


Figura 8. Consumo de Memoria RAM por aplicación desarrollada. Fuente: Elaboración propia

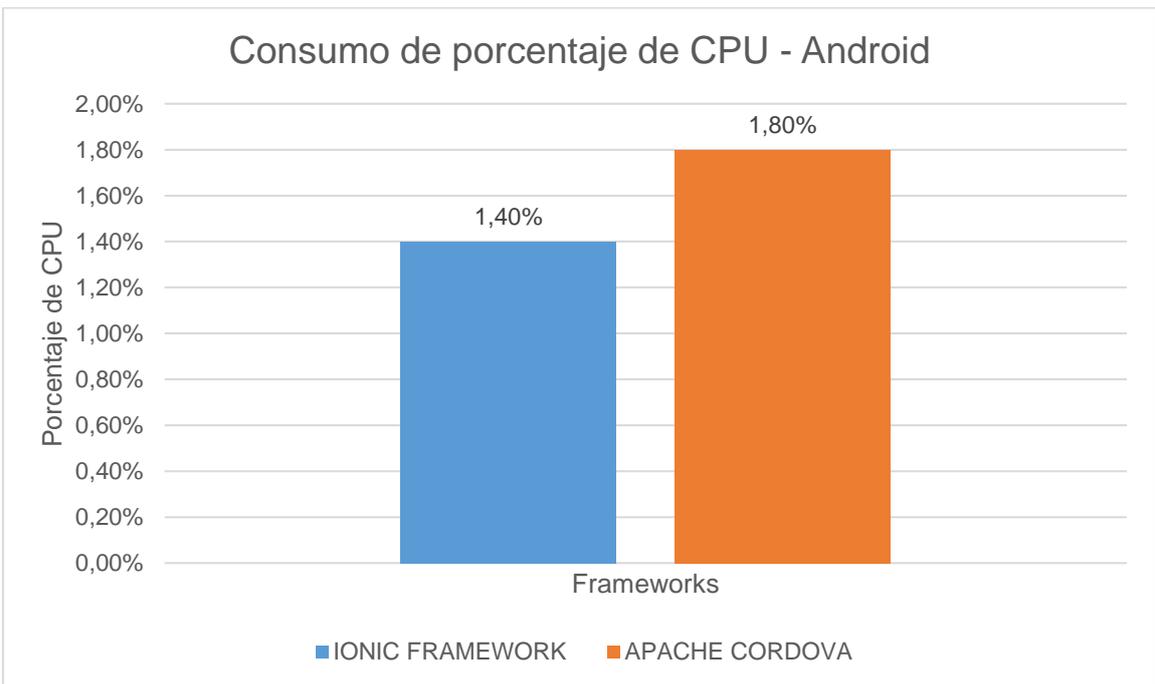


Figura 9. Consumo de porcentaje de CPU por aplicación desarrollada - Android. Fuente: Elaboración propia.

Tabla 14.

Consumo de recursos de PC por aplicación desarrollada - IOS

Framework	Consumo de RAM (Megabytes)	Porcentaje de CPU
IONIC FRAMEWORK	55.70 MB	1.5%
APACHE CORDOVA	62.45 MB	1.7%

*Nota:* La computadora donde se realizaron las pruebas tiene un procesador con las siguientes características: 11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80GHz 2.80 GHz. Fuente: Elaboración propia.

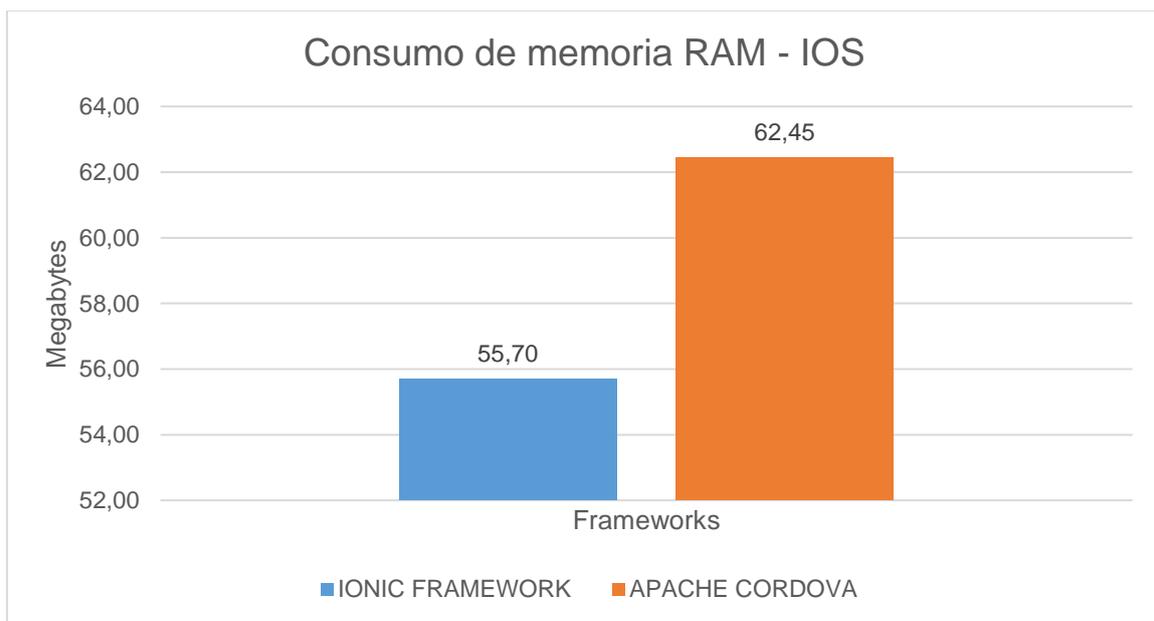
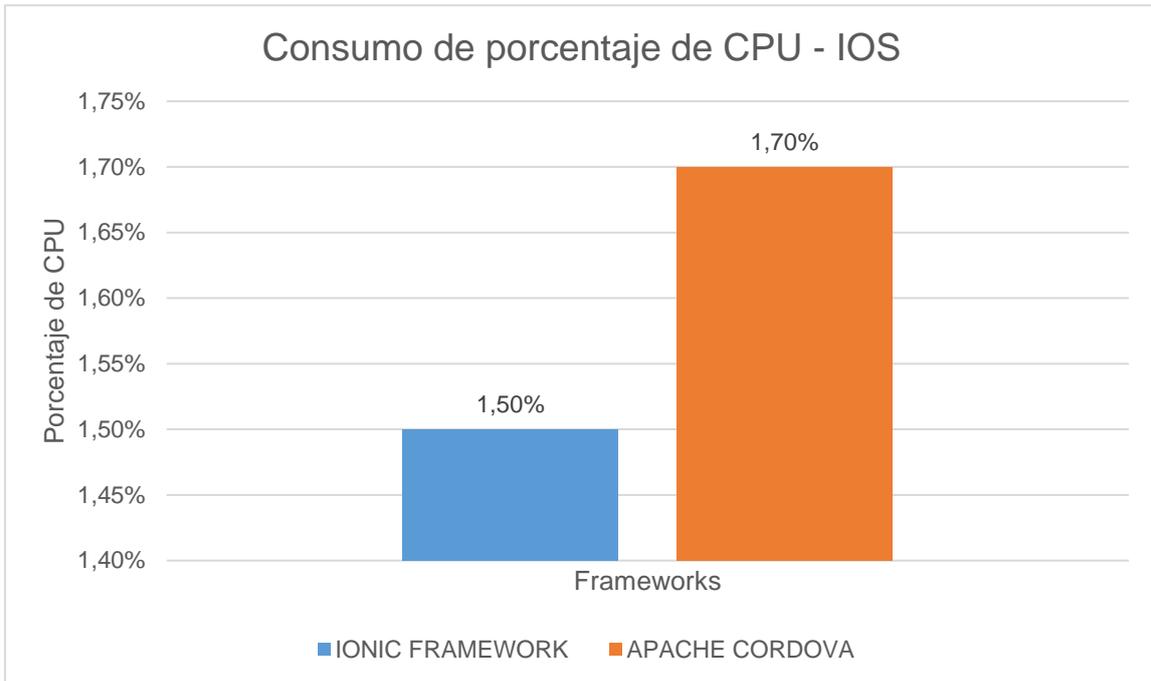


Figura 10. Consumo de Memoria RAM por aplicación desarrollada - IOS. Fuente: Elaboración propia



*Figura 11.* Consumo de porcentaje de CPU por aplicación desarrollada.  
Fuente: Elaboración propia.

Se puede visualizar que como resultados se tiene que la aplicación demo con el CRUD para gestión de usuarios realizada con Ionic Framework consume ligeramente menos recursos que su competidor Apache Cordova, tanto para la plataforma Android como IOS, así mismo la diferencia entre ambos es mínima.

Se tiene en cuenta que la aplicación se integró con cada framework y para cada plataforma (Android e IOS) a fin de tener los resultados más claros y convincentes a la hora de poder medir estos resultados.

Indicar también que la aplicación cuenta con las operaciones CRUD para la gestión de usuarios y se ha desarrollado a modo de prueba para poder tener claridad al momento de realizar a comparativa y análisis respectivo a los frameworks,

- b. Tiempo de respuesta:** Se mide el tiempo de respuesta de cada proceso de las operaciones CRUD implementado en la aplicación móvil según el framework seleccionado. Se ha utilizado la herramienta Jmeter con el

respectivo plan de pruebas.

Tabla 15.  
*Tiempo de respuesta por operación CRUD - Android*

Framework	Create	Read	Update	Delete
IONIC FRAMEWORK	783.16 ms.	669.42 ms.	732.82 ms.	692.96 ms.
APACHE CORDOVA	747.98 ms.	656.82 ms.	721.72 ms.	710.58 ms.

*Nota:* Los tiempos de respuesta fueron medidos en Milisegundos. Fuente: Elaboración propia.

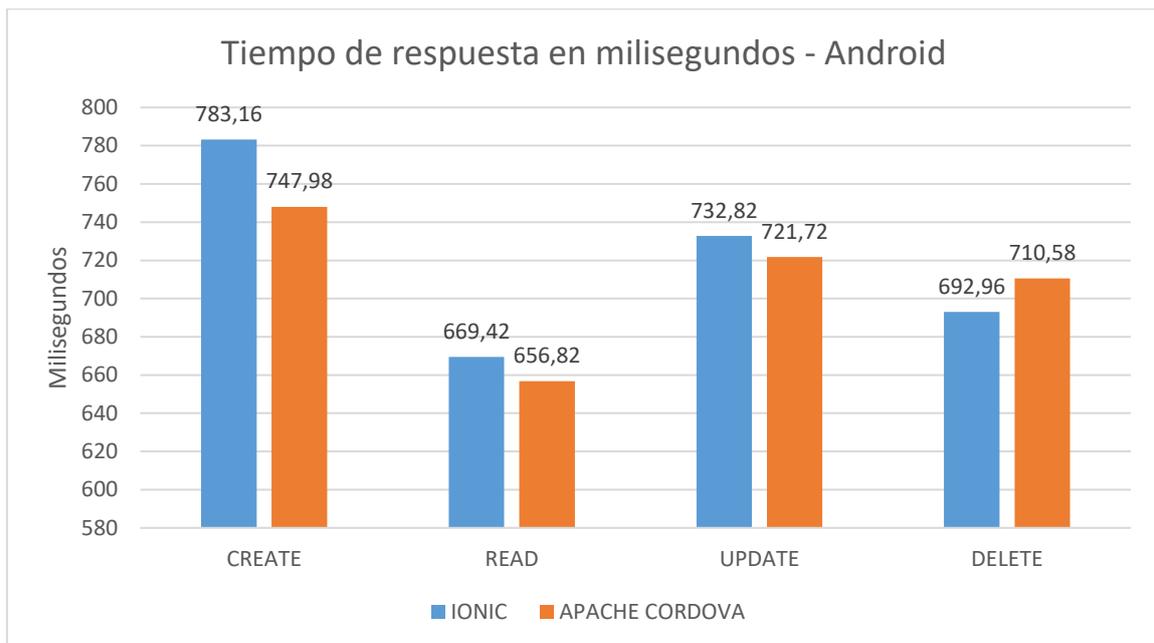


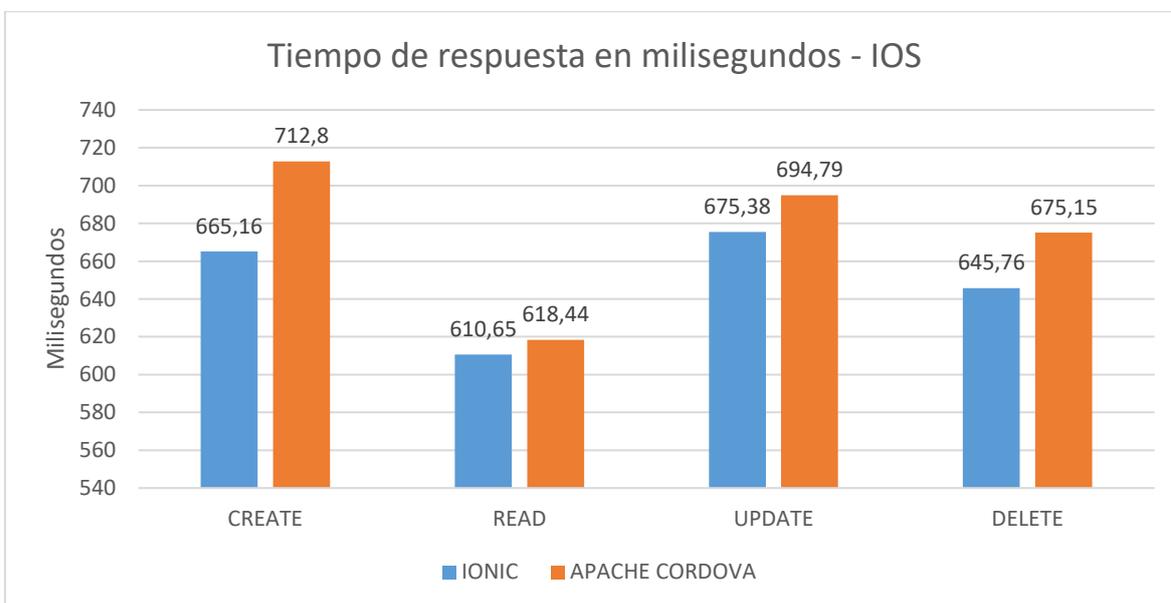
Figura 12. Tiempos de respuestas desglosado por cada una de las operaciones CRUD. Fuente: Elaboración propia.

Tabla 16.  
*Tiempo de respuesta por operación CRUD - IOS*

Framework	Create	Read	Update	Delete
-----------	--------	------	--------	--------

IONIC FRAMEWORK	665.16 ms.	610.65 ms.	675.38 ms.	645.76 ms.
APACHE CORDOVA	712.80 ms.	618.44 ms.	694.79 ms.	675.15 ms.

*Nota:* Los tiempos de respuesta fueron medidos en Milisegundos. Fuente: Elaboración propia.



*Figura 13.* Tiempos de respuestas desglosado por cada una de las operaciones CRUD. Fuente: Elaboración propia.

Se puede evidenciar en el gráfico presentado como resultados en esta sección que la aplicación integrada con Apache Cordova tiene los tiempos de respuesta más cortos con respecto a su competidor Ionic Framework considerando la plataforma Android pero para IOS Ionic Framework tiene tiempos de respuesta más cortos, cabe recalcar que como en anteriores resultados presentados, la diferencia entre ambos es mínima y muchas veces dependerá de las características propias del equipo, así como la cantidad de aplicaciones o programas que se estén ejecutando en simultáneo.

Finalmente indicar que la aplicación desarrollada cuenta con las operaciones CRUD para la gestión de usuarios que fueron implementadas con cada uno de los frameworks y desplegados tanto para Android y IOS a nivel de prueba, ya que se pretende medir los frameworks como tal.

#### **4.2. Discusión de resultados**

Apoyo la premisa del estudio actual basándonos en los datos, que muestran que ni uno ni otro se predice como el mejor, ya que cada uno tiene su propio marco y características, así como su propio conjunto de beneficios e inconvenientes. Mediante el plan de pruebas ejecutado se puede evidenciar que para desarrollar un CRUD con cada uno de los frameworks como parte de la muestra (Ionic Framework y Apache Cordova) no se muestran grandes diferencias entre uno y otro, el desarrollo del CRUD se realizó sin mayores problemas siguiendo la documentación encontrada en cada uno de los sitios oficiales de los frameworks.

Así mismo indicar que para el uso de plantillas para crear aplicaciones móviles fue mucho más fácil la integración con Ionic, pero para implementar con Apache Cordova sólo fue necesario integrar cualquier proyecto web realizado anteriormente dentro de la estructura correspondiente y generar el compilado de la aplicación para obtener resultados.

Por otro lado, estos hallazgos corroboran lo que afirma Suárez, E. (2011): "A través de la construcción de prototipos se reconocieron las fortalezas y deficiencias teóricas y prácticas de cada uno de ellos". Esto es consistente con los hallazgos de esta investigación.

De igual manera, los hallazgos corroboran la conclusión de Sánchez, C. (2015) de que "Si hay poca diferencia entre los marcos, el arquitecto de software elegirá qué marco utilizar en función del proyecto en cuestión". Por último, se aconseja que antes de seleccionar un marco de trabajo, se considere la experiencia de cada persona que va a interactuar con el marco de trabajo a lo largo del desarrollo del proyecto "".

Sin embargo, discrepa de la investigación de Casiano (2019), al afirmar que "el framework Ionic ofrece un mayor grado de eficiencia con un valor de 2,87 en la

plataforma Android y de 2,85 en la plataforma iOS, lo que lo convierte en el framework más idóneo para desarrollar apps móviles multiplataforma."

### **4.3. Aporte práctico**

#### **Propuesta de investigación**

Como parte de la propuesta de investigación se plantea una identificar los mejores frameworks para aplicaciones móviles híbridas existentes actualmente en el mercado, mediante revisión de documentación y artículos de investigación correspondiente. Luego de identificar los frameworks para la muestra de la investigación y aplicando el diseño de investigación cuasi experimental se pueden seleccionar Ionic Framework y Apache Cordova, los mismos que son tomados como muestra para la presente investigación.

Una vez seleccionada la muestra se realiza una aplicación móvil con cada uno de los frameworks que incluye un CRUD de operaciones para administrar usuarios, las operaciones CRUD son un proceso central y principal para casi todas las aplicaciones y sistemas transaccionales, motivo por el cual se ha definido su implementación dentro de la aplicación. Para el desarrollo se propone implementar servicios web con java springboot y una base de datos embebida como parte de la arquitectura propuesta; de manera independiente al desarrollo como tal se propone evaluar y medir la experiencia del desarrollador con cada uno de los frameworks al momento de la implementación del proyecto y la integración con las interfaces de usuario y servicios web.

Finalmente, luego de desarrollar las aplicaciones móviles con cada uno de los frameworks tomados como muestra para la investigación, se inician las pruebas correspondientes para evaluar tiempos de respuesta, consumo de recursos de PC, así como la experiencia del desarrollador en cuanto a la implementación con cada uno, tomando en cuenta las horas de esfuerzo al momento del desarrollo, curva de aprendizaje, acceso a documentación e instalación de los frameworks.

Considerando los objetivos específicos de la presente investigación, se detalla a continuación el procedimiento realizado para cumplir con cada uno de los objetivos planteados inicialmente.

## **OE1: Seleccionar los frameworks para realizar la comparativa.**

En la actualidad existe mucha información relevante en la web sobre análisis comparativos y rankings de los frameworks más usados, los mejores frameworks y los más populares, si bien es cierto en la mayoría de sitios y artículos encontrados los resultados son similares, existe la poca confiabilidad ya que estos sitios carecen de información con fuentes fidedignas, de manera de que no se puede tomar como referencia para la presente investigación, así mismo durante el proceso de la realización del trabajo de investigación para lograr cumplir este objetivo fue necesario investigar diferentes fuentes de información confiables para lograr obtener los frameworks más usados actualmente en el mercado, durante la investigación de artículos y diferentes sitios en la web para encontrar la lista top de los frameworks se han tomado como referencia la investigación realizada con G2 GROWD.

Los marcos de desarrollo móvil ofrecen a los desarrolladores una base de funciones genérica que se puede modificar para un software de aplicación móvil específico. Los marcos se pueden dividir en tres categorías: frameworks nativos para el desarrollo de plataformas específicas, frameworks de aplicaciones web móviles y aplicaciones híbridas que combinan las funciones de frameworks de aplicaciones web nativas y móviles. El departamento de TI de una empresa describe los requisitos y selecciona un marco basado en la plataforma de su elección. Los marcos nativos permiten a los desarrolladores utilizar todas las funciones que son nativas en una plataforma móvil (iOS, Android, Windows, Galaxy). Sin embargo, los marcos nativos son específicos de la plataforma y solo pueden crear aplicaciones para una plataforma móvil. Los marcos híbridos no tienen la capacidad de usar toda la funcionalidad de la API nativa, pero aún brindan herramientas para que los desarrolladores accedan a la cámara, la geolocalización y el sistema de notificación de un teléfono. Los marcos de desarrollo móvil a menudo se acceden y modifican a través de plataformas de desarrollo móvil. (G2 Crowd, 2021).

Considerando la fuente de información como lo es G2 GROWD, plataforma de revisión de software confiable que estudia y compara diferentes tecnologías, se

ha considerado para esta investigación el top que muestra en su sitio, de esta manera la plataforma evalúa ciertas cualidades de cada uno de los frameworks para considerar su top correspondiente y los agrupa en categorías como “Contenders”, “Leaders”, “Niche” y “High Performers” denominando la comparativa como “G2 CROWD Grid for Mobile Development Frameworks”.

A continuación, se muestra la figura correspondiente.

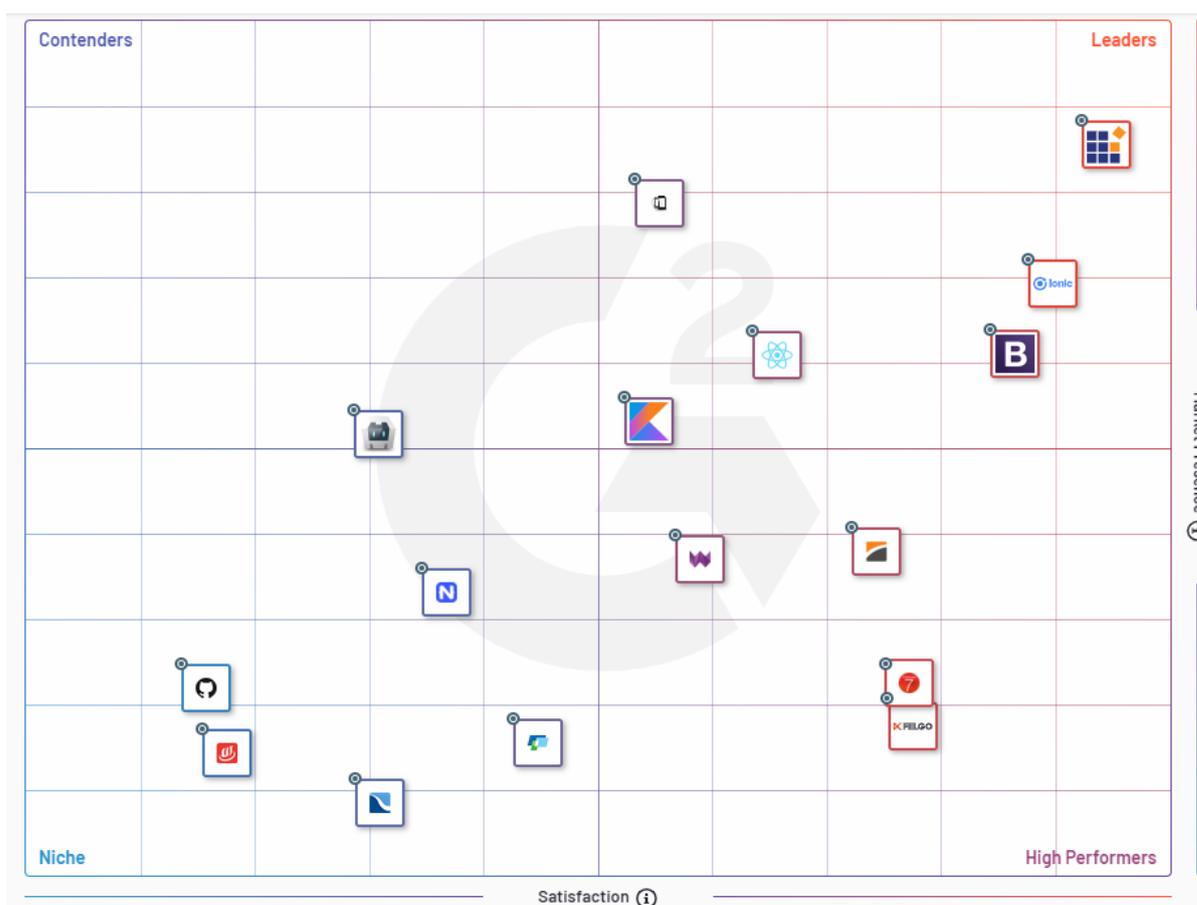


Figura 14. Ranking de los frameworks más usados actualmente. Fuente: G2 CROWD (2021)

Tabla 17.  
Categoría del Top G2 CROWD

Categoría	Descripción
-----------	-------------

Contenders <sup>a</sup>	Con calificaciones por debajo del promedio.
Leaders <sup>a</sup>	Altamente valorados por los usuarios.
Niche <sup>a</sup>	No han recibido suficientes revisiones para su validación
High Performers <sup>a</sup>	Altamente calificados por los usuarios.

Nota: Se describa cada una de las categorías que fueron considerados dentro del estudio y que son mostrados en la figura 11. Tomado de <sup>a</sup>G2 Crowd (2021)

Tabla 18.  
*Criterios del Top G2 CROWD*

<b>Leyenda</b>	<b>Descripción</b>
Satisfaction <sup>a</sup>	Se basa en datos de satisfacción del cliente de opiniones reales.
Market presence <sup>a</sup>	El puntaje de presencia en el mercado de G2, es una combinación de más de 15 puntos de datos de las opiniones de los usuarios.

Nota: Se describa cada uno de los criterios que fueron considerados dentro del estudio y que son mostrados en la figura 11. Tomado de <sup>a</sup>G2 Crowd (2021)

## **OE2: Desarrollar una aplicación como caso de estudio para la comparación.**

Dentro del cumplimiento del este objetivo se propone elaborar una aplicación móvil con la operación CRUD para administrar usuarios con cada uno de los frameworks, se describe los servicios web utilizados, integración con base de datos, integración de proyecto frontend con Ionic Framework y Apache Cordova.

### **A) Propuesta de CRUD**

Las operaciones CRUD (Created, Read, Update, Delete) son un proceso principal en el desarrollo de aplicaciones y sistemas transaccionales, cualquier software transaccional debe tener operaciones CRUD dentro de sus requerimientos funcionales para diferentes entidades del negocio, así mismo lo que se busca dentro de la investigación es obtener el análisis comparativo con cada uno de los frameworks seleccionados como muestra midiendo la

experiencia para el desarrollador y su interacción con las herramientas seleccionadas a nivel de desarrollo, de manera posterior obtener métricas a nivel de aplicación, en este contexto para la investigación se desarrolla una aplicación móvil que contempla la administración de usuarios mediante operaciones CRUD.

Sierra (2017) en su trabajo de fin de grado afirma que la gran mayoría de los proyectos de aplicaciones web que se desarrollan, llevan a cabo numerosas operaciones CRUD, el desarrollo de estas operaciones es muy reiterativo, cada operación suele seguir un patrón que se repite.

## **B) Requerimientos funcionales**

Como parte de los requerimientos funcionales tenemos:

- ✓ Listar usuarios.
- ✓ Crear usuarios.
- ✓ Modificar usuarios.
- ✓ Eliminar usuarios.

## **C) Arquitectura propuesta**

Identificamos la arquitectura propia de cada framework con su detalle correspondiente y en base a ellos se propone la arquitectura a nivel de aplicación para el desarrollo de las operaciones CRUD.

### **Arquitectura Apache Cordova**

Apache Cordova es un marco de trabajo gratuito y de código abierto para el desarrollo de aplicaciones móviles. Permite la programación multiplataforma utilizando tecnologías web estándar del sector como HTML5, CSS3 y JavaScript. Las aplicaciones funcionan en entornos de contenedores optimizados para cada plataforma y hacen uso de enlaces de API compatibles con los estándares para acceder a las características del dispositivo, como sensores, datos y estado de la red (Apache Cordova, 2021).

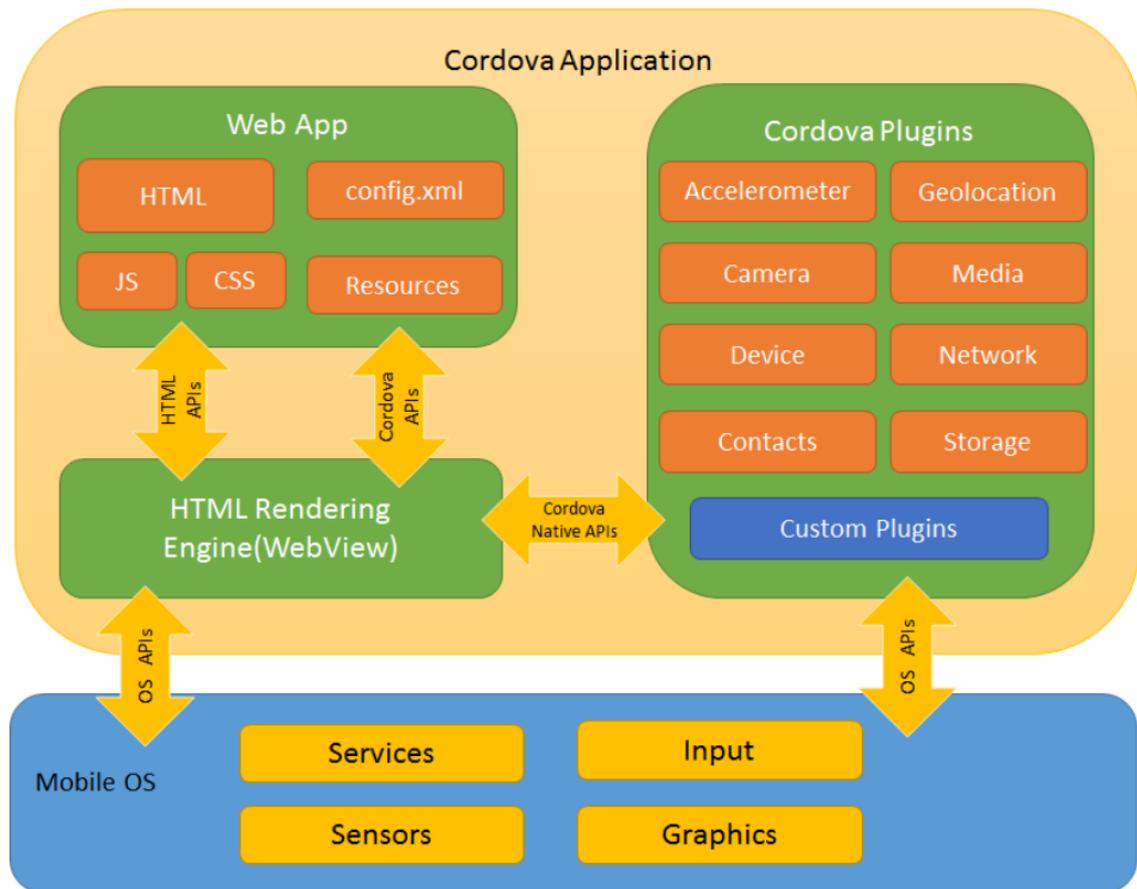


Figura 15. Arquitectura propia del framework Apache Cordova. Fuente Apache Cordova 2021.

### Arquitectura Ionic Framework

Ionic Framework se centra en la interacción y la experiencia de usuario de la interfaz de usuario de una aplicación: Controles de UI, interacciones, gestos y animaciones. Es fácil de aprender y se integra bien con otras bibliotecas y frameworks como Angular, React y Vue. Además, un simple script de inclusión permite utilizarlo independientemente de un marco de trabajo de frontend. (Ionic, 2021).

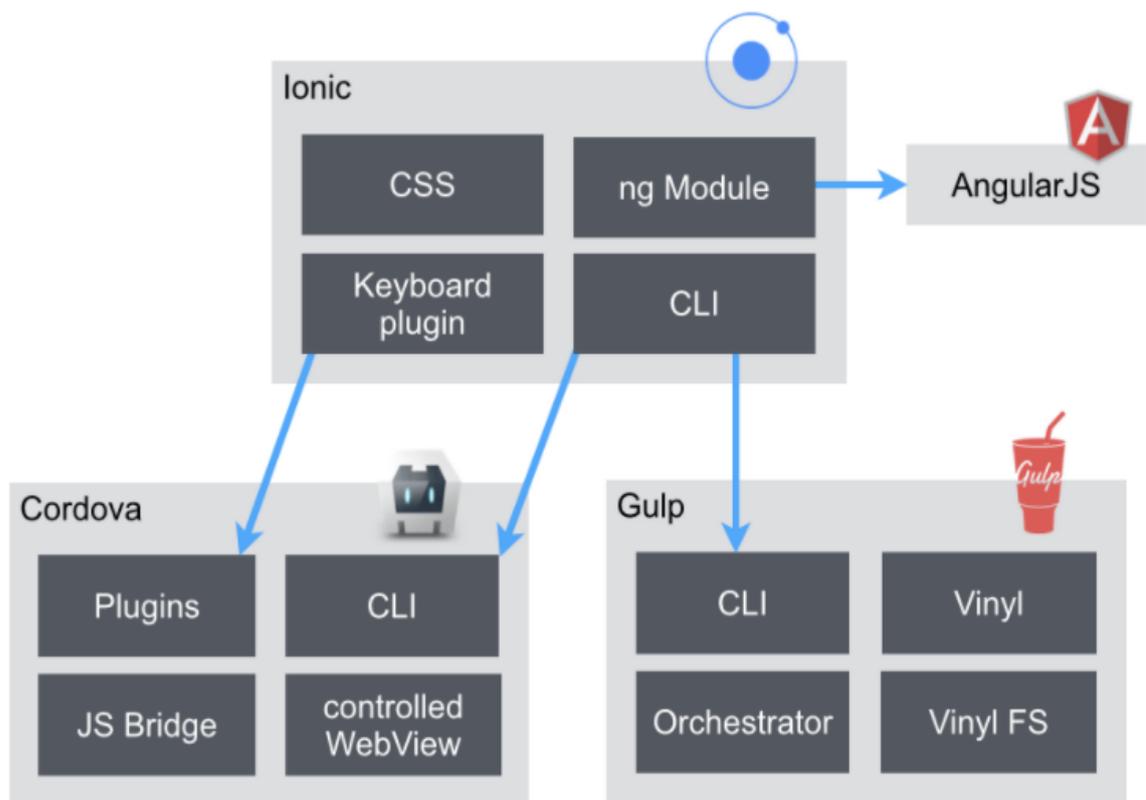
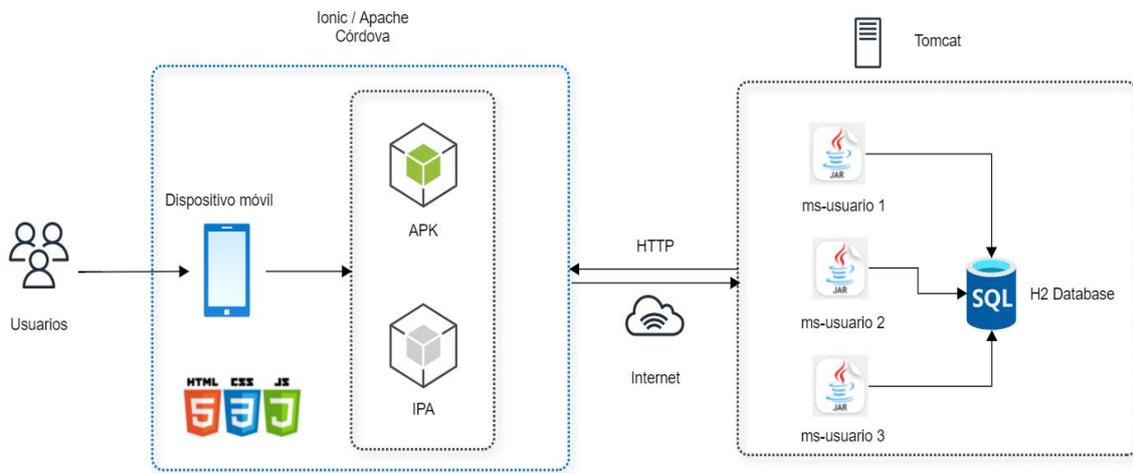


Figura 16. Arquitectura propia que provee Ionic Framework. Fuente: Ionic Framework 2021.

### Arquitectura propuesta para la aplicación móvil.

Se propone la arquitectura orientada a microservicios ya que ésta nos provee una configuración donde cada componente de la aplicación funciona de manera independiente. Estos componentes serán mini aplicaciones que funcionarán por separado y seguirán su propio desarrollo y despliegue.



*Figura 17.* Arquitectura propuesta para el desarrollo de la aplicación móvil.  
Fuente: Elaboración propia.

- ✓ El usuario ingresa al aplicativo mediante su dispositivo móvil.
- ✓ El aplicativo desarrollado con Ionic/Apache Cordova consume servicios web conectándose mediante protocolos HTTP al servidor Tomcat embebido de Springboot que corre dentro de la máquina virtual de java dentro del servidor de aplicaciones.
- ✓ El servidor de aplicaciones expone los microservicios como procesos java.
- ✓ El microservicio se conecta con la base de datos H2 embebida en el servidor Tomcat que provee el ambiente de Springboot.
- ✓ El servidor de aplicaciones responde a las peticiones HTTP que se realizaron desde el aplicativo.
- ✓ Se muestran los resultados en el aplicativo móvil.

La arquitectura sugerida permite la flexibilidad de la aplicación al permitir que los servicios autónomos se diseñen y desplieguen por separado. Además, un error en el servicio no debería tener un efecto adverso en la capacidad de otros servicios para operar de forma independiente y según lo previsto.

Por ejemplo, el microservicio que expone servicios web para la administración

de usuarios es resistente a errores y puede reiniciarse a menudo a efectos de disponibilidad. Esta resiliencia también está relacionada con el estado que se ha almacenado en nombre del microservicio, la facilidad con la que se puede recuperar el estado del microservicio y la capacidad del microservicio para reanudarlo con éxito. En otras palabras, el diseño debe proporcionar resiliencia de proceso (los procesos pueden reanudarse en cualquier momento), así como resiliencia de estado o de datos (no hay pérdida de datos y se mantiene la consistencia de los mismos).

#### D) Base de datos

Para el desarrollo de la aplicación móvil en la presente investigación fue necesario conectarnos a una base de datos para poder persistir los datos ingresados para de esta manera poder realizar las operaciones CRUD para la administración de usuarios. Para este caso se hizo uso del motor de base de datos H2 que viene embebido en el ambiente que provee el servidor Tomcat de SpringBoot, debido a los requerimientos funcionales para el desarrollo del CRUD solo fue necesario crear una sola tabla.

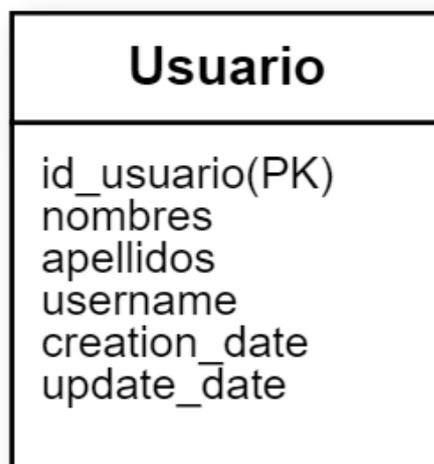


Figura 18. Tabla de usuarios desarrollada por el H2. Fuente: Elaboración propia.

El motor de base de datos que provee SpringBoot se puede visualizar al

levantar el servidor Tomcat y accediendo mediante un navegador web colocando la ruta: <http://localhost:8080/h2-console>

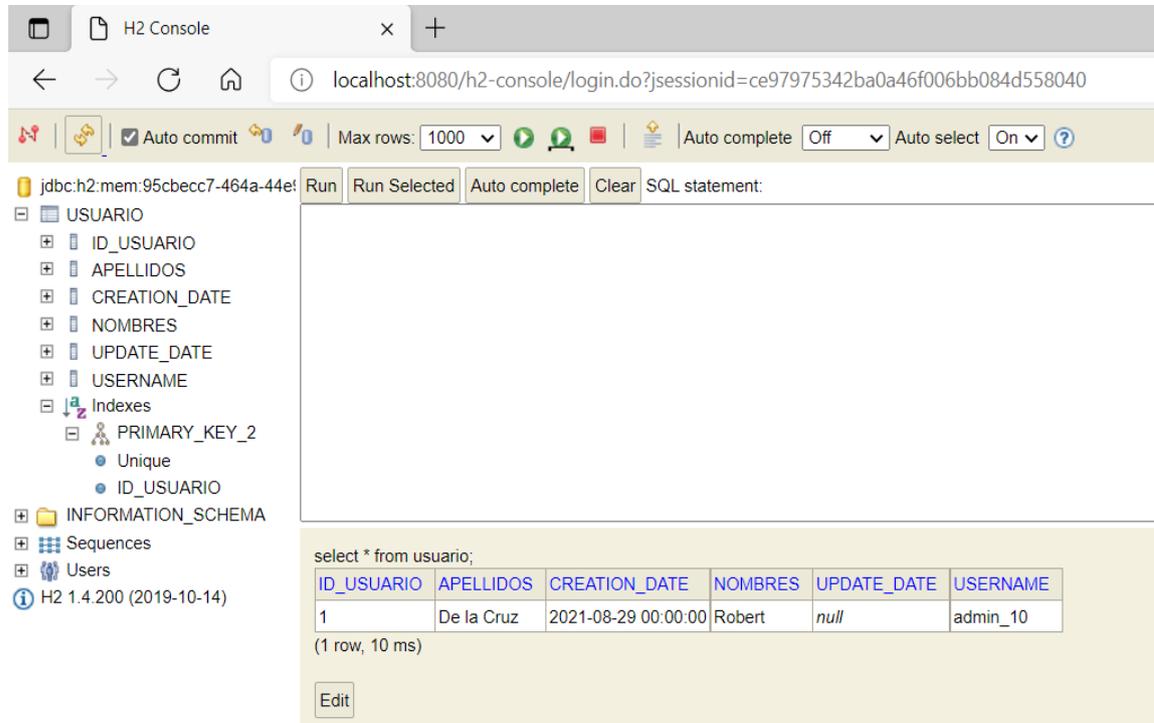


Figura 19. Consola de administración para el motor de base de datos H2.  
Fuente: Elaboración propia.

Tabla 19.  
Diccionario de datos

Tabla	Atributo	Descripción	PK	Tipo de dato
Usuario	id_usuario	Identificador único por usuario	Sí	Integer
	nombres	Nombres del usuario	No	Varchar (255)
	apellidos	Apellidos del usuario	No	Varchar (255)
	username	Username del usuario	No	Varchar (50)
	creation_date	Fecha de creación	No	Timestamp
	update_date	Fecha de actualización	No	Timestamp

Nota: Se describe el diccionario de datos con la información de cada uno los

campos para la tabla “Usuario”. Fuente: Elaboración propia.

## E) Servicios web

Haciendo uso de Java y Springboot, se crearon servicios web para realizar las operaciones CRUD de manera que la aplicación móvil no pueda consumir los servicios y persistir la información ingresada.

Se creó un proyecto que pueda alojar los servicios de manera que estos puedan ser expuestos mediante el servidor Tomcat embebido que provee Springboot y haciendo uso de la arquitectura de microservicios se exponen los servicios web.

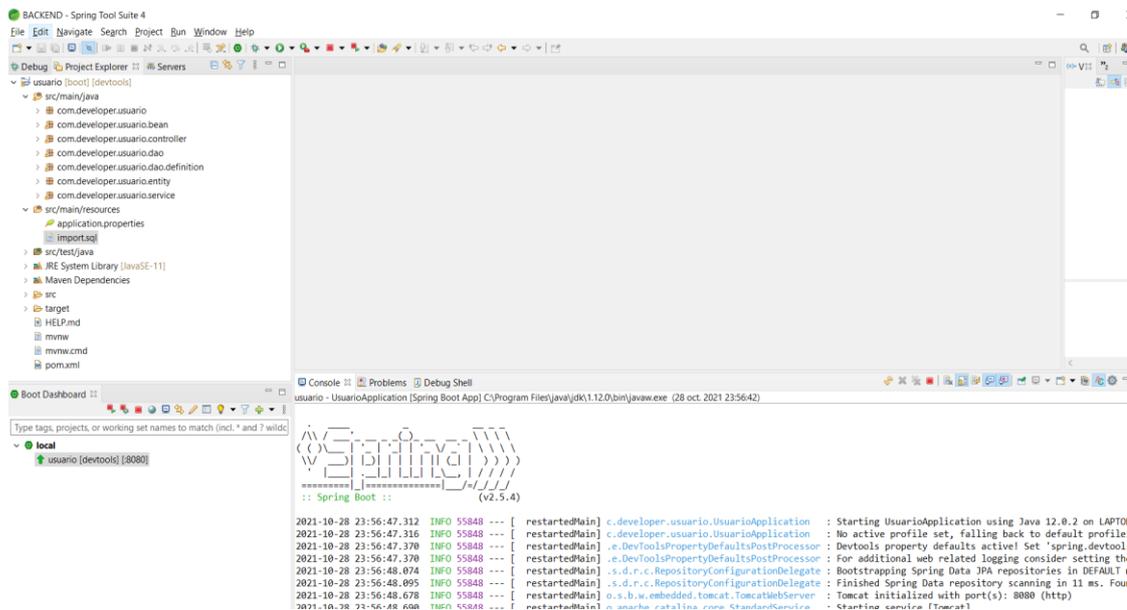
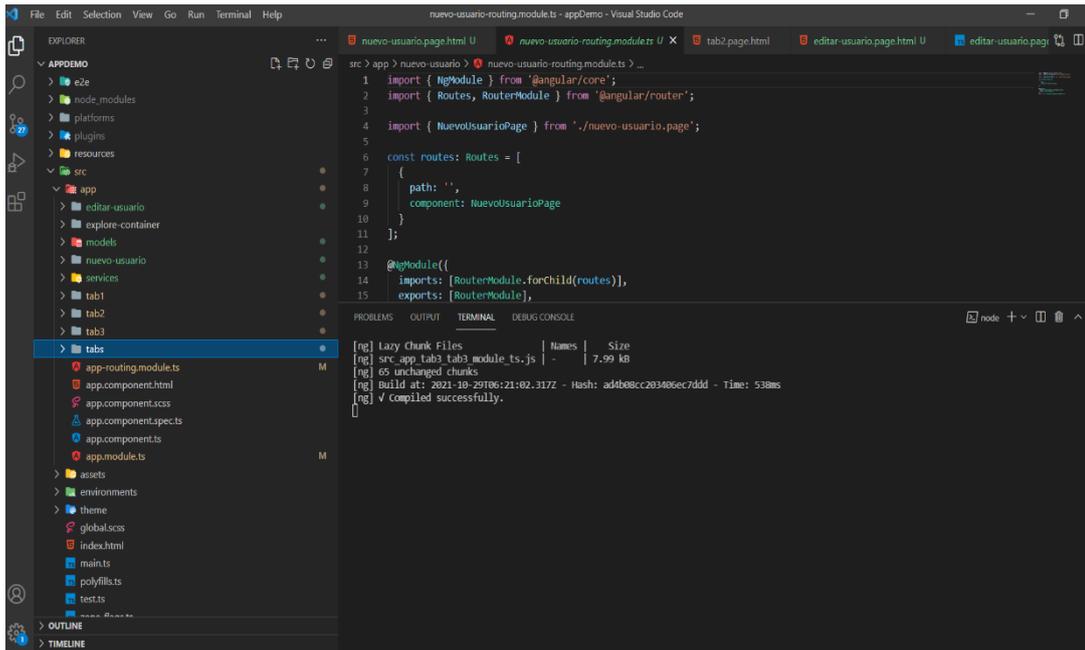


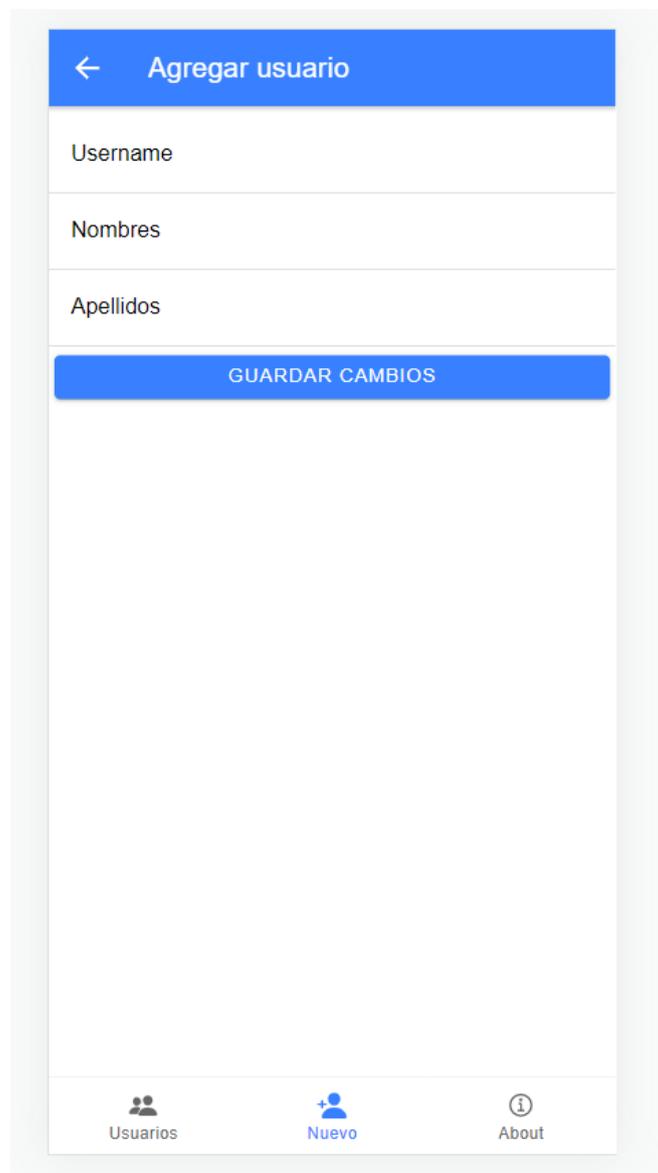
Figura 20. Captura de pantalla que muestra el entorno de desarrollo para los servicios web con Java Springboot. Fuente: Elaboración propia.

## F) Entorno de aplicación móvil

Para el desarrollo de la aplicación móvil fue necesario crear un proyecto frontend que involucra archivos HTML, CSS y JavaScript, para posteriormente hacer la integración tanto con Ionic, así como con Apache Cordova.

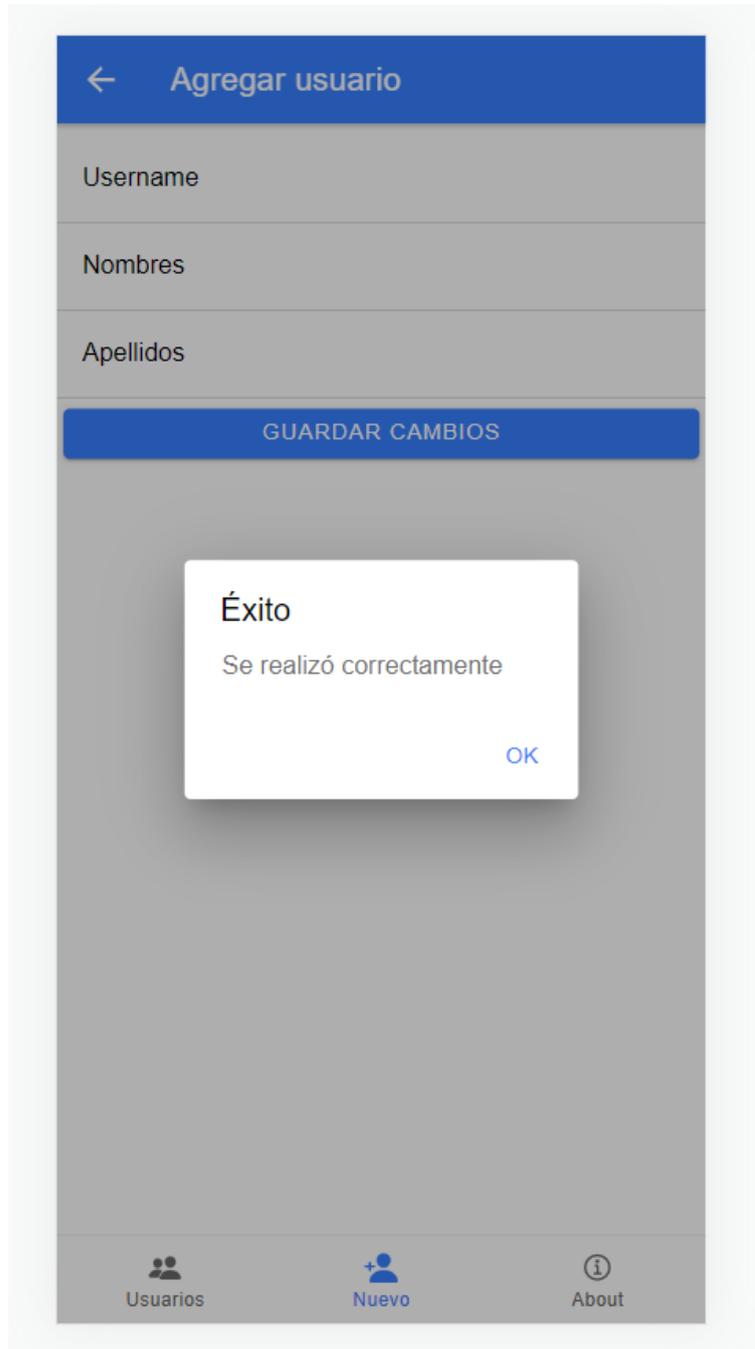
Se presenta a continuación el entorno de desarrollo a nivel frontend.





*Figura 22.* Interfaz gráfica para agregar crear nuevos usuarios. Fuente: Elaboración propia.

De manera posterior a llenar los datos del usuario a registrar, el usuario presiona el botón “Guardar cambios” entonces, la aplicación hace del servicio que registra al usuario mediante una petición HTTP y graba el nuevo registro en la base de datos, la aplicación muestra un mensaje exitoso.

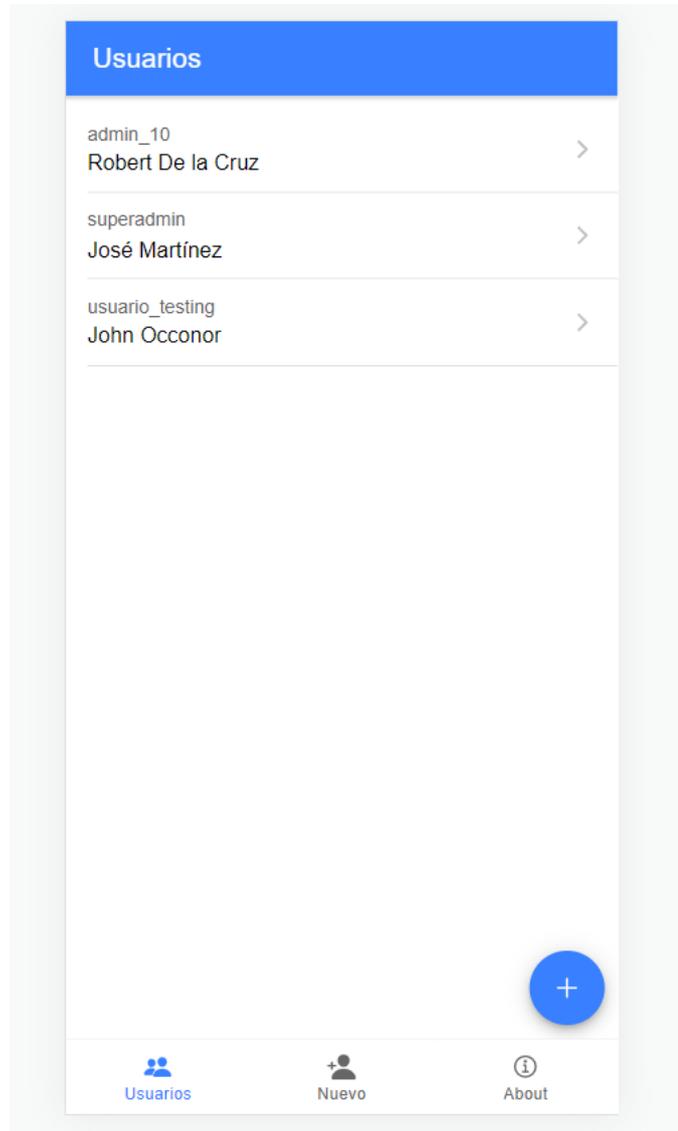


*Figura 23.* Interfaz gráfica que muestra el proceso de creación de usuarios exitoso. Fuente: Elaboración propia.

### **Listar Usuarios**

Como para de los requerimientos funcionales se desea implementar en la aplicación la funcionalidad de “Listar usuarios”, los mismos que fueron agregados mediante la aplicación móvil y el consumo de los servicios web

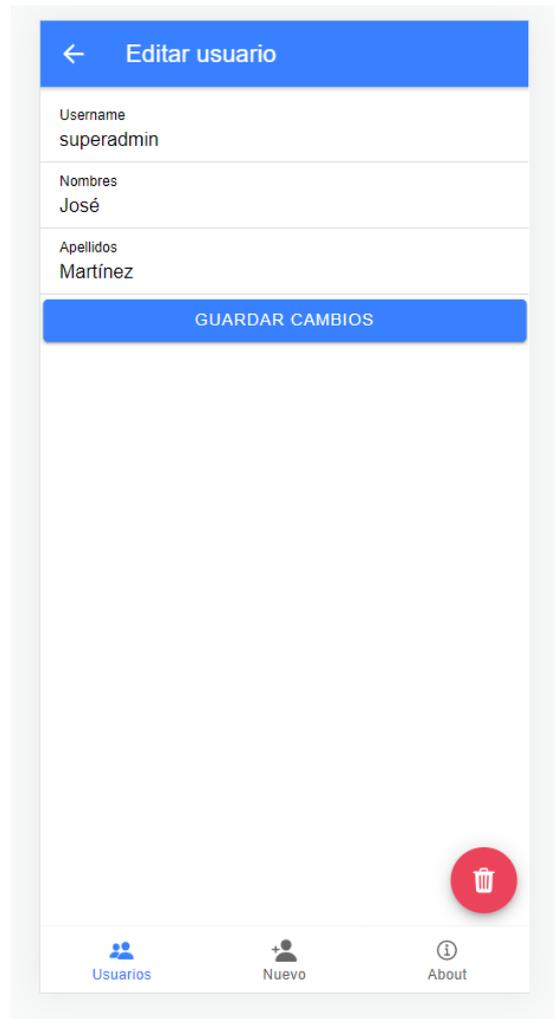
implementados.



*Figura 24.* Interfaz gráfica que lista lo usuarios creados por la aplicación. Fuente: Elaboración propia.

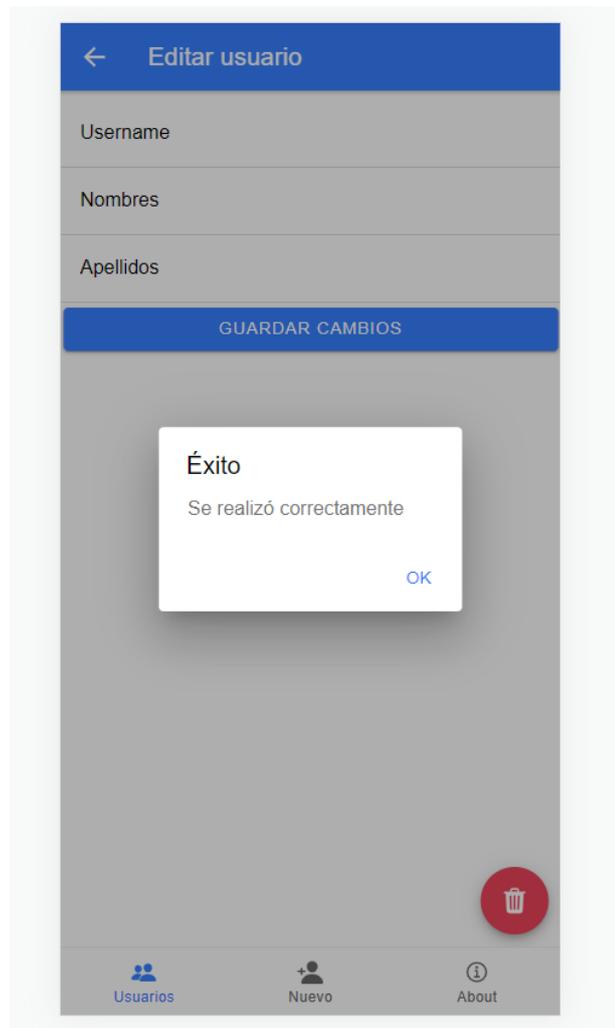
## Editar Usuarios

La aplicación permitirá realizar la actualización de los datos del usuario, basta con presionar sobre el nombre del usuario en la lista de usuarios creados para poder obtener la información del usuario a actualizar.



*Figura 25.* Interfaz gráfica que muestra edición de usuarios. Fuente: Elaboración propia.

Luego de llenar los campos correspondientes, la aplicación guarda el nuevo registro en la base de datos y muestra el mensaje de éxito.

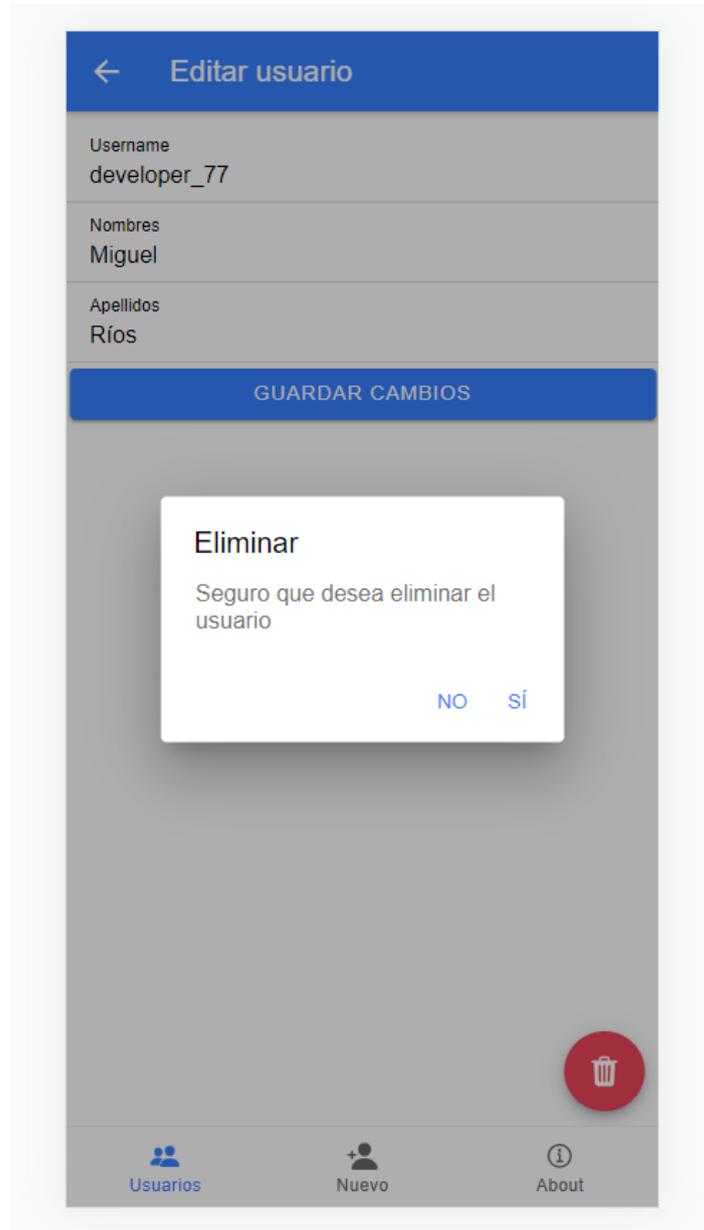


*Figura 26.* Interfaz gráfica que muestra el mensaje satisfactorio luego de realizar la edición de un usuario. Fuente: Elaboración propia.

### **Eliminar Usuarios**

Finalmente, para concluir con los requerimientos funcionales para la aplicación móvil se desea poder eliminar usuarios. El usuario podrá seleccionar el registro a eliminar tocando el nombre en la lista de usuarios, la aplicación lo llevará a la ventana de edición y en la parte inferior se puede visualizar un botón de color rojo el cual permitirá haciendo uso de los servicios web creados anteriormente se procede a realizar la eliminación de un registro.

La aplicación requiere confirmación para proceder con la eliminación.



*Figura 27.* Interfaz gráfica que muestra el mensaje de confirmación antes de realizar la eliminación del usuario seleccionado. Fuente: Elaboración propia.

El usuario presiona sobre el botón sí y la aplicación procede a eliminar el registro de la base de datos y muestra un mensaje de éxito.

### **OE3: Realizar pruebas a las aplicaciones desarrolladas con base en los frameworks**

Para realizar las pruebas correspondientes a las aplicaciones desarrolladas con

cada uno de los frameworks se utilizó la herramienta Jmeter, implementando un plan de pruebas considerando una concurrencia de 50 usuarios para cada una de las operaciones CRUD, de esta forma se obtiene las métricas tanto en consumo de memoria RAM, consumo de porcentaje de CPU y como también medir el tiempo de respuesta por cada petición a los servicios web, luego obtenido los resultados se calcula la media por cada una de las tareas realizadas.

Para realizar las pruebas se ha utilizado una laptop como emulador de un dispositivo móvil, el equipo donde se realizaron las pruebas tiene un procesador con las siguientes características: 11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80GHz 2.80 GHz.

### Consumo de memoria RAM

Esta medición tiene como finalidad poder identificar cuanta memoria RAM consume la aplicación móvil desarrollada tanto para la plataforma Android así como en IOS en cada una de las tareas que se consideraron en el plan de pruebas, de manera posterior se obtiene la media por cada una de las tareas y finalmente se llega a una media que engloba todas las operaciones CRUD. Se muestra a continuación las tablas correspondientes al plan de pruebas ejecutado.

Tabla 20.

*Consumo de memoria RAM por cada una de las operaciones CRUD dentro del plan de pruebas realizado - Ionic Framework - Android*

N° de Tarea	Create (MB) <sup>a</sup>	Read (MB) <sup>a</sup>	Update (MB) <sup>a</sup>	Delete (MB) <sup>a</sup>
1	40	43	65	52
2	46	49	47	34
3	66	45	67	54
4	48	51	73	60
5	63	66	56	67
6	53	56	78	65
7	27	30	52	39
8	40	43	65	52
9	62	65	59	46
10	42	45	67	54
11	62	65	43	30
12	68	57	53	40
13	69	56	55	42
14	68	71	63	65
15	47	50	57	44
16	58	61	60	47

17	69	62	47	34
18	71	58	52	39
19	61	56	49	36
20	52	55	56	43
21	62	65	66	53
22	62	65	47	34
23	31	34	56	43
24	42	45	67	54
25	47	50	72	59
26	58	61	57	44
27	39	42	53	40
28	71	67	59	46
29	61	64	62	49
30	67	48	53	40
31	41	44	46	33
32	42	45	48	35
33	35	38	54	41
34	26	29	51	38
35	55	58	67	54
36	37	40	62	49
37	49	52	56	43
38	34	40	62	49
39	25	28	50	37
40	34	37	52	39
41	42	45	62	49
42	44	48	61	48
43	39	42	64	51
44	30	35	58	45
45	54	48	44	31
46	62	56	43	30
47	67	44	48	35
48	56	59	54	41
49	42	67	45	39
50	42	50	45	32

*Nota:* Los valores mostrados corresponden a la cantidad de <sup>a</sup> Megabytes de memoria RAM que consumen cada uno de las operaciones CRUD. Fuente: Elaboración propia.

Tabla 21.

*Consumo de memoria RAM en promedio con base en el total de pruebas realizadas - Ionic Framework - Android*

	<b>Create (MB)<sup>a</sup></b>	<b>Read (MB)<sup>a</sup></b>	<b>Update (MB)<sup>a</sup></b>	<b>Delete (MB)<sup>a</sup></b>	<b>CRUD(MB)<sup>a</sup></b>
<b>Total</b>	50.16	50.6	56.56	44.48	50.45

*Nota:* Los valores mostrados corresponden a la cantidad de <sup>a</sup> Megabytes total de

memoria RAM que consumen cada uno de las operaciones CRUD. En la columna “CRUD” se muestra el promedio de consumo por Aplicación. Fuente: Elaboración propia.

Tabla 22.

*Consumo de memoria RAM por cada una de las operaciones CRUD dentro del plan de pruebas realizado - Ionic Framework - IOS*

<b>N° de Tarea</b>	<b>Create (MB)<sup>a</sup></b>	<b>Read (MB)<sup>a</sup></b>	<b>Update (MB)<sup>a</sup></b>	<b>Delete (MB)<sup>a</sup></b>
1	55	58	62	49
2	56	59	55	48
3	60	48	58	45
4	55	58	65	52
5	68	71	52	67
6	50	53	62	49
7	53	56	69	56
8	45	48	65	52
9	66	69	59	56
10	62	65	65	52
11	62	65	56	52
12	66	58	54	58
13	72	55	58	44
14	66	69	63	65
15	54	57	55	42
16	68	71	60	47
17	69	62	54	41
18	75	58	52	39
19	63	56	49	36
20	58	61	56	43
21	74	77	66	53
22	68	71	47	34
23	48	51	66	53
24	49	52	68	55
25	50	53	68	55
26	55	58	57	44
27	53	56	53	40
28	68	67	59	46
29	66	69	60	47
30	68	48	53	40
31	51	54	46	33
32	50	53	48	35
33	55	58	54	41
34	46	65	65	52
35	52	55	67	54
36	48	51	65	52
37	58	61	56	43
38	52	45	67	54

39	45	48	66	53
40	49	54	52	54
41	40	48	62	49
42	44	50	61	48
43	59	62	68	55
44	50	40	58	58
45	56	50	44	46
46	68	56	55	48
47	64	48	65	53
48	58	61	54	48
49	52	65	55	52
50	65	55	58	58

*Nota:* Los valores mostrados corresponden a la cantidad de <sup>a</sup> Megabytes de memoria RAM que consumen cada uno de las operaciones CRUD. Fuente: Elaboración propia.

Tabla 23.

*Consumo de memoria RAM en promedio con base en el total de pruebas realizadas - Ionic Framework – IOS*

	<b>Create (MB)<sup>a</sup></b>	<b>Read (MB)<sup>a</sup></b>	<b>Update (MB)<sup>a</sup></b>	<b>Delete (MB)<sup>a</sup></b>	<b>CRUD(MB)<sup>a</sup></b>
<b>Total</b>	57.68	57.56	56.64	44.92	55.70

*Nota:* Los valores mostrados corresponden a la cantidad de <sup>a</sup> Megabytes total de memoria RAM que consumen cada uno de las operaciones CRUD. En la columna “CRUD” se muestra el promedio de consumo por Aplicación. Fuente: Elaboración propia.

Tabla 24.

*Consumo de memoria RAM por cada una de las operaciones CRUD dentro del plan de pruebas – Apache Cordova - Android*

<b>N° de Tarea</b>	<b>Create (MB)<sup>a</sup></b>	<b>Read (MB)<sup>a</sup></b>	<b>Update (MB)<sup>a</sup></b>	<b>Delete (MB)<sup>a</sup></b>
1	50	53	66	53
2	46	69	67	58
3	46	55	77	64
4	48	59	81	68
5	63	66	56	67
6	53	56	78	65
7	57	65	87	74
8	50	53	75	62
9	62	65	59	52
10	42	45	67	69
11	62	65	43	55
12	68	71	53	54

13	69	72	55	42
14	68	71	63	65
15	49	52	57	65
16	48	51	60	57
17	69	72	63	56
18	71	74	52	65
19	61	64	49	56
20	52	55	56	54
21	68	71	66	53
22	62	65	47	55
23	55	58	80	67
24	57	60	82	69
25	47	50	72	59
26	58	61	57	64
27	59	62	53	60
28	71	74	59	66
29	61	64	62	59
30	67	70	53	56
31	61	54	66	54
32	42	55	48	55
33	65	58	54	53
34	46	54	76	63
35	55	56	67	54
36	57	60	82	69
37	59	62	56	63
38	58	60	82	69
39	55	58	69	56
40	64	67	52	65
41	62	65	62	64
42	56	59	61	67
43	59	62	84	71
44	60	60	68	65
45	54	56	65	50
46	62	65	54	50
47	67	70	56	55
48	58	60	58	61
49	49	52	65	52
50	60	63	58	55

*Nota:* Los valores mostrados corresponden a la cantidad de <sup>a</sup> Megabytes de memoria RAM que consumen cada uno de las operaciones CRUD. Fuente: Elaboración propia.

Tabla 25.

*Consumo de memoria RAM en promedio con base en el total de pruebas realizadas - Apache Cordova - Android*

	<b>Create (MB)<sup>a</sup></b>	<b>Read (MB)<sup>a</sup></b>	<b>Update (MB)<sup>a</sup></b>	<b>Delete (MB)<sup>a</sup></b>	<b>CRUD(MB)<sup>a</sup></b>
<b>Total</b>	57.76	61.28	63.56	60	60.65

*Nota:* Los valores mostrados corresponden a la cantidad de <sup>a</sup> Megabytes total de memoria RAM que consumen cada uno de las operaciones CRUD. En la columna “CRUD” se muestra el promedio de consumo por Aplicación. Fuente: Elaboración propia.

Tabla 26.

*Consumo de memoria RAM por cada una de las operaciones CRUD dentro del plan de pruebas – Apache Cordova - IOS*

<b>N° de Tarea</b>	<b>Create (MB)<sup>a</sup></b>	<b>Read (MB)<sup>a</sup></b>	<b>Update (MB)<sup>a</sup></b>	<b>Delete (MB)<sup>a</sup></b>
1	60	65	54	65
2	62	66	65	56
3	65	55	67	67
4	58	59	66	65
5	60	63	56	65
6	58	61	68	65
7	62	65	68	65
8	65	68	66	62
9	63	66	59	69
10	50	53	65	65
11	63	66	66	64
12	65	68	53	55
13	70	73	55	65
14	68	71	63	65
15	55	58	57	75
16	58	61	60	67
17	65	68	66	62
18	56	59	47	64
19	68	71	68	66
20	50	53	66	67
21	66	69	66	53
22	60	63	47	58
23	58	61	68	55
24	55	58	70	66
25	50	53	75	69
26	58	61	57	55
27	59	62	66	54
28	71	74	65	64
29	61	64	68	65
30	65	68	60	56
31	62	54	66	54
32	65	55	68	55
33	64	58	68	65

34	54	54	76	58
35	65	56	66	54
36	55	58	66	65
37	66	69	66	65
38	63	60	68	68
39	68	71	67	66
40	62	65	56	65
41	62	65	66	56
42	65	65	64	67
43	60	63	57	64
44	60	60	53	65
45	55	56	59	66
46	62	65	62	68
47	66	69	53	62
48	58	65	66	60
49	64	67	65	62
50	65	65	58	62

*Nota:* Los valores mostrados corresponden a la cantidad de <sup>a</sup> Megabytes de memoria RAM que consumen cada uno de las operaciones CRUD. Fuente: Elaboración propia.

Tabla 27.

*Consumo de memoria RAM en promedio con base en el total de pruebas realizadas - Apache Cordova - IOS*

	<b>Create (MB)<sup>a</sup></b>	<b>Read (MB)<sup>a</sup></b>	<b>Update (MB)<sup>a</sup></b>	<b>Delete (MB)<sup>a</sup></b>	<b>CRUD(MB)<sup>a</sup></b>
<b>Total</b>	61.30	62.84	62.94	62.72	62.45

*Nota:* Los valores mostrados corresponden a la cantidad de <sup>a</sup> Megabytes total de memoria RAM que consumen cada uno de las operaciones CRUD. En la columna “CRUD” se muestra el promedio de consumo por Aplicación. Fuente: Elaboración propia.

### **Consumo de porcentaje de CPU**

Para obtener este valor se ha medido el porcentaje de CPU que utiliza cada una de las tareas que se tienen dentro del plan de pruebas, logrando capturar el valor por cada una de las operaciones CRUD para que luego se pueda tener la media en consumo de CPU por cada una de las aplicaciones desarrolladas por cada uno de los frameworks para cada una de las plataformas Android y IOS.

Tabla 28.

*Porcentaje de CPU con que consumen las operaciones CRUD, para cada una de*

*las tareas dentro del plan de pruebas – Ionic Framework - Android*

<b>N° de Tarea</b>	<b>Create (%)<sup>a</sup></b>	<b>Read (%)<sup>a</sup></b>	<b>Update (%)<sup>a</sup></b>	<b>Delete (%)<sup>a</sup></b>
1	1.30	1.30	1.40	1.30
2	1.30	1.35	1.40	1.30
3	1.30	1.45	1.25	1.45
4	1.35	1.35	1.35	1.30
5	1.30	1.45	1.55	1.30
6	1.20	1.45	1.25	1.45
7	1.35	1.30	1.55	1.30
8	1.30	1.35	1.55	1.30
9	1.35	1.45	1.45	1.60
10	1.40	1.35	1.25	1.30
11	1.30	1.45	1.25	1.55
12	1.20	1.35	1.45	1.55
13	1.30	1.45	1.25	1.35
14	1.55	1.30	1.60	1.30
15	1.35	1.35	1.25	1.30
16	1.50	1.50	1.55	1.40
17	1.35	1.55	1.35	1.30
18	1.35	1.35	1.45	1.55
19	1.35	1.40	1.25	1.30
20	1.35	1.35	1.35	1.30
21	1.25	1.45	1.40	1.30
22	1.35	1.45	1.55	1.45
23	1.55	1.35	1.30	1.55
24	1.20	1.50	1.45	1.30
25	1.20	1.35	1.55	1.35
26	1.20	1.35	1.45	1.30
27	1.35	1.45	1.55	1.55
28	1.55	1.45	1.45	1.35
29	1.55	1.55	1.25	1.55
30	1.35	1.45	1.45	1.30
31	1.35	1.45	1.55	1.45
32	1.25	1.40	1.35	1.30
33	1.55	1.45	1.35	1.35
34	1.55	1.55	1.25	1.30
35	1.55	1.45	1.25	1.30
36	1.35	1.45	1.45	1.55
37	1.35	1.45	1.45	1.30
38	1.30	1.35	1.35	1.30
39	1.20	1.35	1.55	1.55
40	1.35	1.40	1.45	1.55
41	1.35	1.45	1.35	1.55
42	1.40	1.35	1.60	1.30
43	1.55	1.55	1.35	1.45
44	1.55	1.35	1.35	1.55
45	1.35	1.50	1.60	1.55

46	1.55	1.45	1.55	1.35
47	1.30	1.45	1.45	1.30
48	1.55	1.45	1.45	1.30
49	1.35	1.55	1.55	1.40
50	1.35	1.35	1.45	1.55

*Nota:* Los valores mostrados corresponden al <sup>a</sup> porcentaje del procesador de CPU que consumen cada una de las tareas con operaciones CRUD. Fuente: Elaboración propia

Tabla 29.

*Porcentaje de consumo de CPU en promedio para cada una de las operaciones CRUD - Ionic Framework - Android*

	<b>Create (%)<sup>a</sup></b>	<b>Read (%)<sup>a</sup></b>	<b>Update (%)<sup>a</sup></b>	<b>Delete (%)<sup>a</sup></b>	<b>CRUD (%)<sup>a</sup></b>
<b>Total</b>	1.37	1.42	1.42	1.40	1.4

*Nota:* Los valores mostrados corresponden al <sup>a</sup> porcentaje del procesador de CPU que consumen cada uno de las operaciones CRUD. En la columna “CRUD” se muestra el valor final de consumo por Aplicación. Fuente: Elaboración propia.

Tabla 30.

*Porcentaje de CPU con que consumen las operaciones CRUD, para cada una de las tareas dentro del plan de pruebas – Ionic Framework - IOS*

<b>N° de Tarea</b>	<b>Create (%)<sup>a</sup></b>	<b>Read (%)<sup>a</sup></b>	<b>Update (%)<sup>a</sup></b>	<b>Delete (%)<sup>a</sup></b>
1	1.40	1.40	1.40	1.30
2	1.40	1.40	1.40	1.30
3	1.40	1.45	1.40	1.35
4	1.35	1.45	1.40	1.35
5	1.35	1.40	1.40	1.35
6	1.35	1.40	1.40	1.35
7	1.40	1.45	1.40	1.35
8	1.40	1.45	1.40	1.35
9	1.40	1.45	1.45	1.35
10	1.45	1.45	1.45	1.40
11	1.45	1.45	1.45	1.40
12	1.45	1.45	1.45	1.40
13	1.45	1.45	1.45	1.40
14	1.45	1.50	1.45	1.40
15	1.45	1.50	1.45	1.45
16	1.45	1.50	1.45	1.45
17	1.40	1.50	1.45	1.45
18	1.40	1.50	1.50	1.45
19	1.40	1.50	1.50	1.45
20	1.40	1.50	1.50	1.50
21	1.35	1.50	1.50	1.50

22	1.35	1.55	1.50	1.50
23	1.35	1.55	1.50	1.50
24	1.55	1.50	1.50	1.50
25	1.55	1.55	1.50	1.50
26	1.55	1.55	1.50	1.50
27	1.55	1.50	1.55	1.50
28	1.55	1.50	1.55	1.55
29	1.55	1.50	1.55	1.55
30	1.55	1.50	1.55	1.55
31	1.35	1.50	1.55	1.55
32	1.35	1.55	1.55	1.55
33	1.55	1.55	1.55	1.55
34	1.55	1.55	1.55	1.60
35	1.55	1.60	1.55	1.60
36	1.40	1.60	1.60	1.60
37	1.40	1.55	1.60	1.60
38	1.40	1.55	1.60	1.60
39	1.40	1.60	1.60	1.60
40	1.35	1.60	1.60	1.65
41	1.35	1.55	1.60	1.65
42	1.40	1.55	1.60	1.65
43	1.55	1.65	1.60	1.65
44	1.55	1.65	1.60	1.65
45	1.40	1.65	1.60	1.65
46	1.55	1.65	1.60	1.70
47	1.55	1.65	1.60	1.70
48	1.55	1.65	1.65	1.70
49	1.55	1.65	1.65	1.70
50	1.55	1.65	1.65	1.70

*Nota:* Los valores mostrados corresponden al <sup>a</sup> porcentaje del procesador de CPU que consumen cada una de las tareas con operaciones CRUD. Fuente: Elaboración propia

Tabla 31.

*Porcentaje de consumo de CPU en promedio para cada una de las operaciones CRUD - Ionic Framework - IOS*

	<b>Create (%)<sup>a</sup></b>	<b>Read (%)<sup>a</sup></b>	<b>Update (%)<sup>a</sup></b>	<b>Delete (%)<sup>a</sup></b>	<b>CRUD (%)<sup>a</sup></b>
<b>Total</b>	1.45	1.53	1.52	1.51	1.50

*Nota:* Los valores mostrados corresponden al <sup>a</sup> porcentaje del procesador de CPU que consumen cada uno de las operaciones CRUD. En la columna “CRUD” se muestra el valor final de consumo por Aplicación. Fuente: Elaboración propia.

Tabla 32.

*Porcentaje de CPU con que consumen las operaciones CRUD, para cada una de las tareas dentro del plan de pruebas – Apache Cordova - Android*

<b>N° de Tarea</b>	<b>Create (%)<sup>a</sup></b>	<b>Read (%)<sup>a</sup></b>	<b>Update (%)<sup>a</sup></b>	<b>Delete (%)<sup>a</sup></b>
1	1.95	1.95	1.95	1.55
2	1.90	1.85	1.60	1.55
3	1.30	2.10	1.60	1.45
4	1.85	1.95	1.85	1.30
5	1.90	2.10	1.55	1.55
6	1.85	1.95	1.90	1.85
7	1.85	1.85	1.95	1.55
8	1.85	1.95	1.55	1.55
9	1.85	1.45	1.85	1.55
10	1.90	1.95	1.85	1.85
11	1.90	2.10	1.85	1.55
12	1.95	1.95	1.45	1.55
13	1.85	1.45	1.95	1.85
14	1.55	1.85	1.60	1.65
15	1.85	2.10	1.75	1.85
16	1.50	2.05	1.55	1.40
17	1.85	1.55	1.85	1.65
18	1.85	1.85	1.85	1.55
19	1.85	2.15	1.85	1.65
20	1.85	1.85	1.85	1.85
21	2.15	1.85	1.85	1.85
22	1.85	2.15	1.95	1.85
23	1.55	1.95	1.85	1.55
24	1.20	1.80	1.45	1.75
25	1.95	2.05	1.55	1.75
26	1.90	2.05	1.95	1.65
27	1.90	1.85	1.55	1.55
28	1.55	1.85	1.45	1.95
29	1.55	1.95	1.95	1.55
30	1.85	2.05	1.45	1.95
31	1.90	2.05	1.55	1.95
32	1.25	1.95	1.85	1.65
33	1.55	1.95	1.85	1.75
34	1.55	1.55	1.95	1.75
35	1.55	1.85	1.75	1.95
36	1.95	2.10	1.75	1.85
37	1.95	1.95	1.95	1.85
38	1.85	2.05	1.85	1.95
39	1.85	1.95	1.55	1.55
40	1.85	2.05	1.85	1.75
41	1.85	1.95	1.75	1.95
42	1.95	1.85	1.85	1.65
43	1.55	1.55	1.85	1.85

44	1.95	1.85	1.85	1.75
45	1.85	1.95	1.95	1.85
46	1.95	1.50	1.85	1.95
47	1.90	1.95	1.75	1.85
48	2.05	2.05	1.95	1.95
49	1.85	2.05	1.75	1.85
50	1.90	1.85	1.75	1.95

*Nota:* Los valores mostrados corresponden al <sup>a</sup> porcentaje del procesador de CPU que consumen cada una de las tareas con operaciones CRUD. Fuente: Elaboración propia.

Tabla 33.  
*Porcentaje de consumo de CPU en promedio para cada una de las operaciones CRUD – Apache Cordova - Android*

	<b>Create (%)<sup>a</sup></b>	<b>Read (%)<sup>a</sup></b>	<b>Update (%)<sup>a</sup></b>	<b>Delete (%)<sup>a</sup></b>	<b>CRUD (%)<sup>a</sup></b>
<b>Total</b>	1.79	1.91	1.77	1.73	1.8

*Nota:* Los valores mostrados corresponden al <sup>a</sup> porcentaje del procesador de CPU que consumen cada uno de las operaciones CRUD. En la columna “CRUD” se muestra el valor final de consumo por Aplicación. Fuente: Elaboración propia.

Tabla 34.  
*Porcentaje de CPU con que consumen las operaciones CRUD, para cada una de las tareas dentro del plan de pruebas – Apache Cordova - IOS*

<b>N° de Tarea</b>	<b>Create (%)<sup>a</sup></b>	<b>Read (%)<sup>a</sup></b>	<b>Update (%)<sup>a</sup></b>	<b>Delete (%)<sup>a</sup></b>
1	1.65	1.60	1.60	1.50
2	1.65	1.60	1.60	1.50
3	1.65	1.65	1.60	1.50
4	1.65	1.65	1.60	1.55
5	1.60	1.65	1.60	1.55
6	1.60	1.65	1.60	1.50
7	1.65	1.65	1.60	1.55
8	1.65	1.65	1.65	1.55
9	1.65	1.65	1.65	1.55
10	1.65	1.65	1.65	1.55
11	1.65	1.65	1.65	1.55
12	1.65	1.65	1.65	1.60
13	1.65	1.65	1.65	1.60
14	1.70	1.70	1.70	1.60
15	1.70	1.65	1.70	1.60
16	1.70	1.65	1.70	1.60

17	1.70	1.70	1.70	1.60
18	1.70	1.65	1.70	1.60
19	1.70	1.65	1.70	1.60
20	1.70	1.65	1.75	1.60
21	1.70	1.70	1.75	1.55
22	1.70	1.70	1.75	1.55
23	1.75	1.70	1.75	1.55
24	1.75	1.70	1.75	1.55
25	1.75	1.70	1.75	1.55
26	1.75	1.70	1.75	1.55
27	1.75	1.70	1.80	1.60
28	1.75	1.70	1.80	1.60
29	1.75	1.70	1.80	1.60
30	1.75	1.70	1.80	1.60
31	1.75	1.70	1.80	1.60
32	1.75	1.70	1.80	1.60
33	1.75	1.70	1.85	1.60
34	1.80	1.70	1.85	1.60
35	1.80	1.70	1.85	1.60
36	1.80	1.75	1.85	1.60
37	1.80	1.75	1.85	1.60
38	1.80	1.75	1.85	1.60
39	1.80	1.75	1.85	1.60
40	1.80	1.75	1.85	1.60
41	1.80	1.75	1.90	1.65
42	1.85	1.75	1.90	1.65
43	1.85	1.75	1.90	1.65
44	1.85	1.75	1.90	1.65
45	1.85	1.75	1.90	1.65
46	1.85	1.80	1.90	1.65
47	1.85	1.80	1.95	1.65
48	1.90	1.80	1.95	1.65
49	1.90	1.80	1.95	1.70
50	1.90	1.80	1.95	1.70

*Nota:* Los valores mostrados corresponden al <sup>a</sup> porcentaje del procesador de CPU que consumen cada una de las tareas con operaciones CRUD. Fuente: Elaboración propia.

Tabla 35.  
*Porcentaje de consumo de CPU en promedio para cada una de las operaciones CRUD – Apache Cordova - IOS*

	<b>Create (%)<sup>a</sup></b>	<b>Read (%)<sup>a</sup></b>	<b>Update (%)<sup>a</sup></b>	<b>Delete (%)<sup>a</sup></b>	<b>CRUD (%)<sup>a</sup></b>
<b>Total</b>	1.74	1.70	1.77	1.59	1.70

*Nota:* Los valores mostrados corresponden al <sup>a</sup> porcentaje del procesador de CPU

que consumen cada uno de las operaciones CRUD. En la columna “CRUD” se muestra el valor final de consumo por Aplicación. Fuente: Elaboración propia.

### Tiempos de respuesta

Dentro de este plan de pruebas se han considerado los tiempos de respuesta en milisegundos al consumo de los servicios web que ejecutan cada una de las operaciones CRUD para cada una de las aplicaciones desarrolladas tanto para Ionic Framework como para Apache Cordova en las plataformas Android e IOS.

Tabla 36.

*Tiempos de respuesta para cada una de las operaciones CRUD dentro del plan de pruebas – Ionic Framework - Android*

N° de Tarea	Create (ms) <sup>a</sup>	Read (ms) <sup>a</sup>	Update (ms) <sup>a</sup>	Delete (ms) <sup>a</sup>
1	602	519	582	543
2	668	546	609	570
3	646	573	636	597
4	680	650	713	674
5	669	562	625	586
6	635	515	578	539
7	627	510	573	534
8	660	543	606	567
9	692	575	638	599
10	686	569	632	593
11	682	565	628	589
12	694	577	640	601
13	703	586	649	610
14	709	592	655	616
15	705	588	651	612
16	711	594	657	618
17	710	593	656	617
18	716	599	662	623
19	718	601	664	625
20	701	584	647	608
21	716	599	662	623
22	713	596	659	620
23	731	614	677	638
24	742	625	688	649
25	747	630	693	654
26	758	641	704	665
27	769	652	715	676
28	771	654	717	678
29	761	644	707	668
30	767	650	713	674

31	781	664	727	688
32	802	685	748	709
33	795	678	741	702
34	816	699	762	723
35	825	708	771	732
36	852	735	798	759
37	859	742	805	766
38	900	783	846	807
39	915	798	861	822
40	922	805	868	829
41	936	819	882	843
42	944	827	890	851
43	939	822	885	846
44	950	833	896	857
45	956	839	902	863
46	962	845	908	869
47	965	848	911	872
48	991	875	938	899
49	972	847	910	871
50	987	873	956	874

*Nota:* Los valores mostrados corresponden al tiempo de respuesta en <sup>a</sup> milisegundos que consumen cada una de las tareas con operaciones CRUD.  
Fuente: Elaboración propia

Tabla 37.  
*Tiempos de respuesta en promedio cada una de las operaciones CRUD – Ionic Framework - Android*

	<b>Create (ms)<sup>a</sup></b>	<b>Read (ms)<sup>a</sup></b>	<b>Update (ms)<sup>a</sup></b>	<b>Delete (ms)<sup>a</sup></b>
<b>Total</b>	783.16	669.42	732.82	692.96

*Nota:* Los valores mostrados corresponden al promedio de tiempo de respuesta en <sup>a</sup> milisegundos que consumen cada una de las operaciones CRUD. Fuente: Elaboración propia.

Tabla 38.  
*Tiempos de respuesta para cada una de las operaciones CRUD dentro del plan de pruebas – Ionic Framework - IOS*

<b>Nº de Tarea</b>	<b>Create (ms)<sup>a</sup></b>	<b>Read (ms)<sup>a</sup></b>	<b>Update (ms)<sup>a</sup></b>	<b>Delete (ms)<sup>a</sup></b>
1	650	545	645	622
2	655	530	639	628
3	658	543	645	631
4	656	535	600	630

5	662	569	632	616
6	662	565	628	612
7	662	577	640	618
8	667	586	649	617
9	667	592	655	623
10	668	588	651	625
11	666	594	657	618
12	661	577	640	643
13	663	586	649	646
14	669	592	655	656
15	652	588	651	650
16	661	594	657	638
17	665	593	656	647
18	663	599	662	644
19	662	601	664	645
20	654	584	647	638
21	668	599	662	666
22	658	596	659	639
23	652	614	677	645
24	657	620	683	644
25	671	610	673	634
26	668	641	704	665
27	658	652	715	676
28	661	654	710	671
29	655	642	705	666
30	669	650	713	674
31	661	664	727	688
32	662	614	677	638
33	664	625	688	649
34	653	630	693	654
35	658	641	704	665
36	669	632	695	656
37	671	634	697	658
38	673	644	683	648
39	675	630	693	654
40	673	651	704	665
41	678	617	680	641
42	677	625	680	646
43	675	632	684	654
44	676	638	700	661
45	673	642	703	646
46	675	620	683	632
47	673	643	706	652
48	678	631	688	649
49	679	652	715	654
50	675	651	746	651

---

*Nota:* Los valores mostrados corresponden al tiempo de respuesta en <sup>a</sup> milisegundos que consumen cada una de las tareas con operaciones CRUD.

Fuente: Elaboración propia

Tabla 39.

*Tiempos de respuesta en promedio cada una de las operaciones CRUD – Ionic Framework - IOS*

	<b>Create (ms)<sup>a</sup></b>	<b>Read (ms)<sup>a</sup></b>	<b>Update (ms)<sup>a</sup></b>	<b>Delete (ms)<sup>a</sup></b>
<b>Total</b>	665.16	610.65	675.38	645.76

*Nota:* Los valores mostrados corresponden al promedio de tiempo de respuesta en <sup>a</sup> milisegundos que consumen cada una de las operaciones CRUD. Fuente: Elaboración propia.

Tabla 40.

*Tiempos de respuesta para cada una de las operaciones CRUD dentro del plan de pruebas – Apache Cordova - Android*

<b>N° de Tarea</b>	<b>Create (ms)<sup>a</sup></b>	<b>Read (ms)<sup>a</sup></b>	<b>Update (ms)<sup>a</sup></b>	<b>Delete (ms)<sup>a</sup></b>
1	508	506	571	560
2	634	533	598	587
3	612	560	625	614
4	646	637	702	691
5	635	549	614	603
6	601	502	567	556
7	593	497	562	551
8	626	530	595	584
9	658	562	627	616
10	652	557	621	610
11	648	552	617	606
12	660	564	629	618
13	669	573	638	627
14	675	579	644	633
15	671	575	640	629
16	677	581	646	635
17	676	580	645	634
18	682	586	651	640
19	684	588	653	642
20	667	571	636	625
21	682	586	651	640
22	679	583	648	637
23	697	601	666	655
24	708	612	677	666
25	713	617	682	671
26	724	628	693	682

27	735	639	704	693
28	737	641	706	695
29	727	631	696	685
30	733	637	702	691
31	747	651	716	705
32	768	672	737	726
33	761	665	730	719
34	782	686	751	740
35	791	695	760	749
36	818	722	787	776
37	825	729	794	783
38	866	770	835	824
39	881	785	850	839
40	888	792	857	846
41	902	806	871	860
42	910	814	879	868
43	905	819	874	863
44	916	820	885	874
45	922	826	896	880
46	928	833	898	886
47	931	835	901	889
48	958	862	927	916
49	938	842	907	896
50	953	860	925	914

*Nota:* Los valores mostrados corresponden al tiempo de respuesta en <sup>a</sup> milisegundos que consumen cada una de las tareas con operaciones CRUD.  
Fuente: Elaboración propia.

Tabla 41.

*Tiempos de respuesta en promedio cada una de las operaciones CRUD – Apache Cordova – Android.*

	<b>Create (ms)<sup>a</sup></b>	<b>Read (ms)<sup>a</sup></b>	<b>Update (ms)<sup>a</sup></b>	<b>Delete (ms)<sup>a</sup></b>
<b>Total</b>	747.98	656.82	721.72	710.58

*Nota:* Los valores mostrados corresponden al promedio de tiempo de respuesta en <sup>a</sup> milisegundos que consumen cada una de las operaciones CRUD. Fuente: Elaboración propia.

Tabla 42.

*Tiempos de respuesta para cada una de las operaciones CRUD dentro del plan de pruebas – Apache Cordova - IOS*

<b>N° de Tarea</b>	<b>Create (ms)<sup>a</sup></b>	<b>Read (ms)<sup>a</sup></b>	<b>Update (ms)<sup>a</sup></b>	<b>Delete (ms)<sup>a</sup></b>
1	535	510	695	547

2	624	537	607	574
3	635	564	560	601
4	662	641	555	678
5	668	553	588	590
6	664	506	620	543
7	650	501	614	538
8	669	534	610	571
9	655	566	622	603
10	657	561	631	597
11	660	556	610	593
12	662	568	622	605
13	659	577	631	614
14	655	583	637	620
15	651	579	633	616
16	657	585	639	622
17	656	584	638	621
18	662	590	644	627
19	664	592	646	629
20	647	548	629	612
21	662	563	644	627
22	659	560	641	624
23	677	578	659	642
24	688	589	670	653
25	693	594	675	658
26	704	605	617	669
27	722	616	565	680
28	725	618	567	682
29	715	608	557	646
30	691	614	563	652
31	705	618	577	666
32	726	616	598	667
33	719	609	591	680
34	740	630	612	701
35	749	635	621	710
36	730	666	648	737
37	737	673	655	744
38	778	714	666	773
39	793	729	711	763
40	791	713	718	770
41	805	727	731	784
42	813	735	720	792
43	808	741	735	787
44	816	742	746	798
45	815	747	758	804
46	831	744	759	810
47	832	716	762	813
48	831	683	768	810
49	841	623	768	818
50	852	681	756	796

---

*Nota:* Los valores mostrados corresponden al tiempo de respuesta en <sup>a</sup> milisegundos que consumen cada una de las tareas con operaciones CRUD.  
Fuente: Elaboración propia.

Tabla 43.

*Tiempos de respuesta en promedio cada una de las operaciones CRUD – Apache Cordova – IOS.*

	<b>Create (ms)<sup>a</sup></b>	<b>Read (ms)<sup>a</sup></b>	<b>Update (ms)<sup>a</sup></b>	<b>Delete (ms)<sup>a</sup></b>
<b>Total</b>	712.8	618.44	649.79	675.15

*Nota:* Los valores mostrados corresponden al promedio de tiempo de respuesta en <sup>a</sup> milisegundos que consumen cada una de las operaciones CRUD. Fuente: Elaboración propia.

### **Aportes finales**

Los resultados de la investigación denominada “Análisis comparativo de frameworks de desarrollo para aplicaciones móviles híbridas” aporta de manera significativa a la toma de decisiones en equipos de desarrollo, departamentos de TI de empresas, organizaciones e instituciones al momento de elegir, seleccionar el framework que mejor se adapta y acomoda a los requerimientos del proyecto a móvil híbrido a implementar.

El impacto se da en el corto - mediano plazo ya que minimizando esfuerzo al momento de seleccionar un framework para desarrollar aplicaciones móviles híbridas, ayuda a la organización a ahorrar costos en la contratación de programadores de aplicaciones nativas, de tal manera que una aplicación móvil híbrida se podrá deployar tanto para la plataforma Android como IOS.

En muchas ocasiones tanto los equipos de desarrollo de proyectos de software tardan en elegir las tecnologías a utilizar para iniciar con el proyecto, con esta investigación se intenta brindar claridad con respecto al uso de cada uno de los framework que son parte del estudio, optimizando tiempos de desarrollo, tiempos de entrega y presupuesto.

El estudio beneficia a todos los equipos de desarrollo de software aportando conocimiento sobre la curva de aprendizaje, horas de esfuerzo por desarrollo, instalación, documentación, consumo de recursos, etc. con cada uno de los

frameworks que son parte del estudio como son Ionic Framework y Apache Cordova.

Así mismo el trabajo de investigación sirve como guía para posibles posteriores estudios referentes al tema, de modo que este estudio puede significar gran aporte a nuevas investigaciones.

## **V. CONCLUSIONES Y RECOMENDACIONES**

### **6.1. Conclusiones**

Para la presente investigación se han obtenido las siguientes conclusiones.

- ✓ En este trabajo de investigación se realizó el análisis comparativo de los frameworks para el desarrollo de aplicaciones móviles híbridas, se tomó como muestra a Ionic Framework y Apache Cordova, se desarrolló una aplicación móvil con cada uno y se realizaron las pruebas correspondientes al análisis, de manera que se pudo determinar que framework elegir a la hora de iniciar un proyecto de desarrollo de una aplicación móvil híbrida.
- ✓ Se seleccionaron los frameworks para realizar la comparativa considerando la plataforma de comparación de software G2 CROWD como fuente confiable, para de esta manera determinar la población y muestra para la presente investigación. Se consideraron criterios cuatro grandes grupos de categorías “Contenders”, “Leaders”, “Niche” y “High Performers” denominando la comparativa como “G2 CROWD Grid for Mobile Development Frameworks”.
- ✓ Se desarrolló una aplicación como caso de estudio para la comparación con cada uno de los framework seleccionados como muestra Ionic Framework y Apache Cordova, tomando como referencia operaciones CRUD, ya que se considera que estas operaciones son básicas para cualquier desarrollo de aplicaciones y sistemas transaccionales, así mismo se consideró relevante la experiencia del desarrollador con cada

uno de los frameworks teniendo en cuenta la curva de aprendizaje, acceso a tutoriales y documentación.

- ✓ Se realizaron pruebas necesarias a las aplicaciones desarrolladas con base en los frameworks tomados como muestra considerando métricas como tiempo de respuesta, consumo de memoria RAM y consumo de porcentaje de CPU dando como resultados que Apache Cordova es 10.315 milisegundos más rápido que Ionic Framework en promedio como tiempo de respuesta obtenido para la plataforma Android y Ionic Framework es más rápido en promedio por una diferencia de 14.808 milisegundos que Apache Cordova para la plataforma IOS; mientras que en el consumo de memoria RAM Ionic Framework consume 10.20 Megabytes menos de memoria en comparación con Apache Cordova para Android y para la plataforma IOS, Ionic Framework consume 6.75 Megabytes menos que Apache Cordova, finalmente el consumo de porcentaje de CPU Ionic Framework consume 0.4% menos que Apache Cordova para Android y 0.3% menos que Apache Cordova para la plataforma IOS.

## **6.2.Recomendaciones**

A continuación, se presentan algunas recomendaciones para tener en cuenta en trabajos futuros dentro del campo de la ingeniería.

Al momento de seleccionar los frameworks como parte de la muestra de la investigación se recomienda elegirlos considerando que tengan buena documentación en su página oficial, adicionalmente validar la facilidad de encontrar materiales como tutoriales web, así como ejemplos de implementación de aplicaciones demo y la instalación e integración con tecnologías frontend.

Considerar siempre las versiones más estables de los frameworks disponibles dentro de la página oficial de cada uno, muchas veces se lanzan versiones cada cierto periodo de tiempo, pero éstas nuevas versiones no son estables o no pueden ser compatibles con algunas librerías o dependencias y pueden tener algunos errores que ya son corregidos cuando la versión pasa a ser estable.

Con el paso del tiempo es probable que se desarrollen más sistemas operativos para plataformas móviles, por ello se recomienda que los frameworks a utilizar siempre sean compatibles con las nuevas plataformas o sistemas operativos que se encuentren en vigencia.

## REFERENCIAS.

- Apache Cordova (2021). Architecture. Retrieved Aug 12, 2021, from <https://cordova.apache.org/docs/en/10.x/guide/overview/index.html>
- Ionic Framework (2021). Getting started Overview. Aug 12, 2021, <https://ionicframework.com/docs>
- R. Nunkesser, (2018). *Beyond Web/Native/Hybrid: A New Taxonomy for Mobile App Development*. IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft), pp. 214-218.
- V. P. La Manna and F. Pasveer, (2018). *Towards a Framework for Proximity-Based Hybrid Mobile Applications*. IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft), pp. 176-179.
- Y. Yang, Y. Zhang, P. Xia, B. Li y Z. Ren, (2017). *Mobile Terminal Development Plan of Cross-Platform Mobile Application Service Platform Based on Ionic and Cordova*. International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII). pp. 100-103, doi: 10.1109/ICIICII.2017.28.
- C.M. Pinto y C. Coutinho, (2018). *From Native to Cross-platform Hybrid Development*. International Conference on Intelligent Systems (IS), pp. 669-676, doi: 10.1109/IS.2018.8710545.
- I. Malavolta, (2016). *Web-Based Hybrid Mobile Apps: State of the Practice and Research Opportunities*. IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft), pp. 241-242, doi: 10.1109/MobileSoft.2016.050.
- S. Bosnic, I. Papp and S. Novak, (2016). *The development of hybrid mobile applications with Apache Cordova*. 24th Telecommunications Forum (TELFOR), pp. 1-4, doi: 10.1109/TELFOR.2016.7818919.
- B. Mathur y S.M. Satapathy, (2019). *An Analytical Comparison of Mobile Application Development using Agile Methodologies*. 3rd International Conference on Trends in Electronics and Informatics (ICOEI), pp. 1147-1152, doi: 10.1109/ICOEI.2019.8862532.
- K. Vishal y A. S. Kushwaha, (2018). *Mobile Application Development Research Based on Xamarin Platform*. 4th International Conference on Computing Sciences (ICCS), pp. 115-118, doi: 10.1109/ICCS.2018.00027.

- E. Morocho y E. Gómez-Torres, (2019). Application of ISO 25000 for Comparative Analysis of Cross-Platform Mobile Application Development Tools. International Conference on Information Systems and Computer Science (INCISCOS), pp. 319-325, doi: 10.1109/INCISCOS49368.2019.00057.
- P. Que, X. Guo and M. Zhu, (2016). *A Comprehensive Comparison between Hybrid and Native App Paradigms*. 8th International Conference on Computational Intelligence and Communication Networks (CICN) pp. 611-614, doi: 10.1109/CICN.2016.125.
- G2 Crowd. (2021). *Best Mobile Development Frameworks | G2 Crowd*. Retrieved Oct 28, 2021, from <https://www.g2.com/categories/mobile-development-frameworks#grid>
- Inei (2019). Perú: *Tecnologías de Información y Comunicaciones en las Empresas, 2018 – Encuesta Económica Anual 2019*. Retrieved Oct 28, 2021, from [https://www.inei.gob.pe/media/MenuRecursivo/publicaciones\\_digitales/Est/Lib1815/libro.pdf](https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1815/libro.pdf)
- Domínguez, F., Paredes, M., & Santacruz, L. (2014). *Programación multimedia y dispositivos móviles*. (S. A. E. y P. A-MA, Ed.) (Editorial). Madrid: RA-MA. Retrieved Oct 28, 2021, from [www.ra-ma.es](http://www.ra-ma.es)
- Molina, Y., Sandoval, J., & Toledo, S. A. (2012). *Sistema Operativo Android: Características y Funcionalidad para dispositivos móviles*. UNIVERSIDAD TECNOLÓGICA DE PEREIRA. Retrieved Oct 31, 2021, from <http://repositorio.utp.edu.co/dspace/bitstream/11059/2687/1/0053M722.pdf>
- Luján Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante, España: Club Universitario.
- Puetate & Ibarra (2020). *Aplicaciones móviles híbridas*. Retrieved Oct 20, 2021, from <https://www.pucesi.edu.ec/webs2/wp-content/uploads/2021/02/Aplicaciones-M%C3%B3viles-H%C3%ADbridas-2020.pdf>
- Cruz, J. (2012). *Programación en Java*. Retrieved Oct 20, 2021, from <https://www.eumed.net/libros-gratis/ciencia/2012/12/que-es-java.html>
- Guncay Barzallo, M. C., & Samaniego Mosquera, J. W. (2016). *Evaluación de frameworks para el desarrollo de aplicaciones móviles multiplataforma*. Retrieved Oct 20, 2021, from

<https://dspace.unl.edu.ec/jspui/handle/123456789/11177>

Pérez, J., & Gardey, A. (2008). Actualizado: 2021.

*Definición de html*. Retrieved Oct 20, 2021, from <https://definicion.de/html>

Varun B. (2021). *Hybrid Mobile App Frameworks You Can Opt for in 2021 & Beyond*.

Retrieved Oct 20, 2021, from <https://www.pixelcrayons.com/blog/the-top-hybrid-mobile-app-frameworks-for-2021>

Casiano, B. J. (2019). Evaluación de frameworks open source para medir la eficiencia en aplicaciones móviles multiplataforma. Universidad Señor de Sipan.

Retrieved Oct 20, 2021 from <http://repositorio.uss.edu.pe/handle/20.500.12802/6296>

Sierra, J. (2017). Una propuesta de automatización de operaciones CRUD en IFML.

Universidad de Extremadura Escuela Politécnica, Retrieved Oct 20, 2021 from [https://dehesa.unex.es:8443/bitstream/10662/7277/1/TFGUEX\\_2017\\_Sierra\\_Blazquez.pdf](https://dehesa.unex.es:8443/bitstream/10662/7277/1/TFGUEX_2017_Sierra_Blazquez.pdf)

Haro, A. & Quishpi, E. G. (2019). *Desarrollo de una aplicación web y móvil híbrida*

*para la gestión y seguimiento de juicios en el estudio jurídico “Morales & Asociados”*. Escuela Superior Politécnica de Chimborazo. Riobamba.

Retrieved Oct 20, 2021 from <http://dspace.epoch.edu.ec/handle/123456789/13653>

Palomino Aldazabal, R. Y. (2018). Aplicación móvil híbrida que gestiona en tiempo real las preferencias de los clientes en los restaurantes del distrito de Andahuaylas,

2018. Universidad Nacional José María Arguedas. Andahuaylas. Retrieved Oct 20, 2021 from <http://repositorio.unajma.edu.pe/handle/123456789/375>

Montes Cornejo, L. A. (2020). Aplicativo móvil para la ubicación e identificación de

Pymes. Universidad Tecnológica del Perú. Lima. Retrieved Oct 20, 2021 from <https://hdl.handle.net/20.500.12867/3973>

Navajas Segovia, M., & Pamplona Roche, S. (2018). Evaluación de la facilidad de aprendizaje de frameworks JavaScript, Backbone, Angular y Ember.

*Dialnet*. Retrieved Oct 20, 2021 from

<https://dialnet.unirioja.es/servlet/articulo?codigo=6775562>

Espinoza Mina, M. A., & Sierra Cedeño, A. Y. (2018). Análisis Comparativo entre

ASP.NET y PHP. *Dialnet*. Retrieved Oct 20, 2021 from

<https://dialnet.unirioja.es/servlet/articulo?codigo=6779622>

Francia Peralta, N. F. (2019). Análisis comparativo de frameworks open source para el nivel de productividad en los proyectos de desarrollo de software web. Universidad Tecnológica del Perú. Retrieved Oct 20, 2021 from <https://hdl.handle.net/20.500.12867/2736>

Castillo Chinchay, F. & Dávila Hurtado, J. A. (2020). Desarrollo de una Aplicación Móvil para gestionar el proceso De registro de ventas, en la empresa Despensa Peruana S.A. Universidad Nacional Pedro Ruiz Gallo. Retrieved Oct 20, 2021 from <https://hdl.handle.net/20.500.12893/8844>

Gamma, E. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.

Bravo Rivera, R. (2021). Análisis comparativo entre frameworks, para el desarrollo de aplicaciones móviles multiplataformas. Universidad Técnica de Babahoyo. Babahoyo. Retrieved Oct 20, 2021 from <http://dspace.utb.edu.ec/handle/49000/9473>

Oros, J.C. (2002). Diseño de páginas web interactivas con Java Script y CSS. (3a.ed.). México: Editorial Alfaomega.

Suárez, E. (2011). Análisis comparativo de los Frameworks EJB3 y ADO.NET Entity framework para el desarrollo de aplicaciones empresariales. Escuela Politécnica Nacional. Quito. Retrieved Oct 20, 2021 from <http://bibdigital.epn.edu.ec/handle/15000/4295>

Guerrero, C., y Recaman, H. (2009). Marcos de Trabajo (Framework) para soportar el desarrollo de aplicaciones Web de código abierto, 10-25, (Sic) Editorial Ltda., Bucaramanga, Colombia. (2009).

Sánchez Acosta, C. A. (2015). Análisis comparativo de frameworks para el desarrollo de aplicaciones web en java. Universidad Señor de Sipán. Chiclayo. Retrieved Oct 20, 2021 from <https://hdl.handle.net/20.500.12802/242>.

Gutierrez, J. (s,f). ¿Qué es un Framework Web? Retrieved Oct 20, 2021 from [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf)

Molina Ríos, J.R., Zea Ordóñez, M.P., Contenido Segarra, M.J. y García Zerda, F.G. (2018). Comparación de metodologías en aplicaciones web. 3C Tecnología: glosas de innovación aplicadas a la pyme, 7(1). 1-19. DOI: <<http://dx.doi.org/10.17993/3ctecno.2018.v7n1e25.1-19/>>

Thomas, P., Delia, L., Corbalan, L., Cáceres, G., Fernández Sosa, J., Tesone, F., Cuitiño, A., Pesado, P. (2018). Tendencias en el desarrollo de Aplicaciones para Dispositivos Móviles. Retrieved Oct 20, 2021 from [https://digital.cic.gba.gob.ar/bitstream/handle/11746/8316/11746\\_8316.pdf?sequence=1&isAllowed=y](https://digital.cic.gba.gob.ar/bitstream/handle/11746/8316/11746_8316.pdf?sequence=1&isAllowed=y)

## ANEXOS

### Anexo 1. Resolución de aprobación del proyecto de investigación



FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO  
RESOLUCIÓN N°0908-2021/FIAU-USS

Pimentel, 7 de octubre 2021

**VISTO:**

El Acta de reunión N°0109-2021, remitida mediante oficio N°0316-2021/FIAU-IS-USS de fecha 13 de septiembre de 2021 por la Escuela profesional de INGENIERÍA DE SISTEMAS, para la ejecución de la Tesis: “ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS DE DESARROLLO PARA APLICACIONES MÓVILES HÍBRIDAS”, presentado por DE LA CRUZ CERON ROBERTO CARLOS, del Programa de estudios de INGENIERÍA DE SISTEMAS, y;

**CONSIDERANDO:**

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: “La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.”;

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: “Los temas de trabajo de investigación, trabajo académico y tesis son aprobados por el Comité de Investigación y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El periodo de vigencia de los mismos será de dos años, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.”

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; es individual o en pares para obtener un título profesional. Asimismo, en su artículo 25° señala: “El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C.”.

Que, mediante documentos de vistos, el Comité de investigación de la referida Escuela profesional acordó aprobar la ampliación de la vigencia del Proyecto de tesis denominado “ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS DE DESARROLLO PARA APLICACIONES MÓVILES HÍBRIDAS”, aprobada mediante Resolución N° 0345-2017/FIAU-USS de fecha 4 de agosto de 2017, a cargo de DE LA CRUZ CERON ROBERTO CARLOS, en condición de egresado, del Programa de estudios de INGENIERÍA DE SISTEMAS.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

**SE RESUELVE:**

**ARTÍCULO ÚNICO: AMPLIAR VIGENCIA**, de la Tesis denominada “ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS DE DESARROLLO PARA APLICACIONES MÓVILES HÍBRIDAS”, a cargo de DE LA CRUZ CERON ROBERTO CARLOS, del Programa de estudios INGENIERÍA DE SISTEMAS, hasta el 31 de diciembre de 2021.

**REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE**

  
  
Mg. Victor Alexei Tuerto Monteza  
Decano (a) / Facultad De Ingeniería,  
Arquitectura Y Urbanismo  
UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.

  
  
MBA. María Noelia Sialer Rivera  
Secretaría Académica / Facultad de Ingeniería,  
Arquitectura y Urbanismo  
UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.

Cc: Interesado, Archivo





## **GUIA DE ANALISIS DOCUMENTAL**

### ***Acerca de los Frameworks de desarrollo de aplicación móviles híbridas***

Tema de interés	Contenidos analizados
Categorías frameworks	Los frameworks se pueden categorizar por su alcance y compatibilidad.
	Para aplicaciones web, de plataformas nativas, de plataformas híbridas.
Definición	Brindan a los desarrolladores una base genérica de funcionalidad que se puede modificar para software específico de aplicaciones móviles
	Proporcionan patrones y estructura ya definidos.
	Estandariza el desarrollo web.
	Facilita la integración con cualquier proyecto que tenga HTML, CSS y JavaScript
Categoría de evaluación	Contenders
	Leaders
	Niche
	High performers.
Criterios de evaluación	Satisfacción de los usuarios.
	Presencia en el mercado.

### Entrevista

**Título de la Investigación:** ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS DE DESARROLLO PARA APLICACIONES MÓVILES HÍBRIDAS

**Objetivo:** Obtener información correspondiente a la curva de aprendizaje en los frameworks Ionic y Apache Córdoba.

**Entrevistado:**

**Cargo:**

**Framework:**

### Cuestionario de Preguntas

1.- Indique en horas esfuerzo cuánto tiempo tomó realizar el servicio web para agregar un usuario:

- A. 1 a 4 horas
- B. 4 a 8 horas
- C. 8 a 16 horas
- D. más de 16 horas

2.- Indique en horas esfuerzo cuánto tiempo tomó implementar las pantallas para agregar un usuario:

- A. 1 a 4 horas
- B. 4 a 8 horas
- C. 8 a 16 horas
- D. más de 16 horas

3.- Indique en horas esfuerzo cuánto tiempo tomó realizar el servicio web para listar usuarios:

- A. 1 a 4 horas
- B. 4 a 8 horas
- C. 8 a 16 horas
- D. más de 16 horas

4.- Indique en horas esfuerzo cuánto tiempo tomó implementar las pantallas para listar usuarios:

- A. 1 a 4 horas
- B. 4 a 8 horas
- C. 8 a 16 horas
- D. más de 16 horas

5.- Indique el grado de dificultad que implica la integración del proyecto al framework seleccionado.

- A. Fácil
- B. Intermedio
- C. Difícil
- D. Muy difícil

6.- Seleccione la alternativa con la que se identifica al momento de buscar documentación, videotutoriales y material didáctico en la web sobre el framework seleccionado.

- A. Fácil - Hay bastante material en la web
- B. Intermedio - Hay poco material en la web
- C. Difícil - Hay muy poco material en la web
- D. No hay material en la web.

7.- En base a su experiencia en el desarrollo con el framework seleccionado a qué nivel de programadores recomendaría utilizar el framework por primera vez?

- A. Programador Junior
- B. Mínimo programador Semi Senior
- C. Mínimo programadores Senior
- D. Todas las anteriores.

8.- Califique su experiencia en general en el proyecto utilizando el framework seleccionado.

- A. Buena
- B. Regular
- C. Mala
- D. Muy mala

9.- Volvería a utilizar el framework seleccionado en otros proyectos de desarrollo de aplicación móvil híbrida?

- A. Sí, definitivamente.
- B. Probablemente sí.
- C. Lo pensaría.
- D. Definitivamente no.



## Entrevista

**Título de la Investigación:** ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS DE DESARROLLO PARA APLICACIONES MÓVILES HÍBRIDAS

**Objetivo:** Obtener información correspondiente al proceso de instalación de los frameworks

**Entrevistado:**

**Cargo:**

**Framework:**

### Questionario de Preguntas

1.- Se encontraron dificultades al momento de encontrar el instalador del framework y su descarga en la web?

- A. Sí
- B. No

2.- Existe documentación en español correspondiente para lograr la descarga e instalación del framework?

- A. Sí
- B. No

3.- El instalador del framework tiene selector de idiomas?

- A. Sí
- B. No

4.- Es posible instalar el framework mediante consola de comandos?

- A. Sí
- B. No

5.- ¿Qué tiempo tardó el proceso de instalación del framework?

- A. Menor a 5 minutos
- B. De 5 a 10 minutos
- C. Supera los 10 minutos

6.- Califique su experiencia en el proceso de instalación del framework

- A. Buena
- B. Regular
- C. Mala
- D. Muy mala

7.- Califique el grado de dificultad sobre el proceso de instalación del framework.

- A. Fácil

- B. Intermedio
- C. Complejo
- D. Avanzado

## GUÍA, JUICIO DE EXPERTOS

### 1. Identificación del Experto

Nombre y Apellidos: ...EDGAR BAJALQUI CARRASCO.....

Centro laboral: .....TATA CONSULTING.....

Título profesional: .....INGENIERO INFORMÁTICO.....

Grado: ..... Mención:.....

Institución donde lo obtuvo: PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ.....

Otros estudios.....

### 2. Instrucciones

Estimado(a) especialista a continuación se muestra un conjunto de indicadores, el cual tienes que evaluar con criterio ético y estrictez científica, la validez del instrumento propuesto (véase anexo N° 1).

Para evaluar dicho instrumento, marca con un aspa(x) una de las categorías contempladas en el cuadro:

1: Inferior al básico 2: Básico 3: Intermedio 4: Sobresaliente 5: Muy sobresaliente

### 3. Juicio de experto

INDICADORES	CATEGORÍA				
	1	2	3	4	5
1. Las dimensiones de la variable responden a un contexto teórico de forma (visión general)				X	
2. Coherencia entre dimensión e indicadores (visión general)					X
3. El número de indicadores, evalúan las dimensiones y por consiguiente la variable seleccionada (visión general)					X
4. Los ítems están redactados en forma clara y precisa, sin ambigüedades (claridad y precisión)					X
5. Los ítems guardan relación con los indicadores de las variables(coherencia)				X	
6. Los ítems han sido redactados teniendo en cuenta la prueba piloto (pertinencia y eficacia)				X	
7. Los ítems han sido redactados teniendo en cuenta la validez de contenido				X	

8. Presenta algunas preguntas distractoras para controlar la contaminación de las respuestas (control de sesgo)					X
9. Los ítems han sido redactados de lo general a lo particular(orden)				X	
10. Los ítems del instrumento, son coherentes en términos de cantidad(extensión)				X	
11. Los ítems no constituyen riesgo para el encuestado(inocuidad)					X
12. Calidad en la redacción de los ítems (visión general)				X	
13. Grado de objetividad del instrumento (visión general)					X
14. Grado de relevancia del instrumento (visión general)				X	
15. Estructura técnica básica del instrumento (organización)				X	
<b>Puntaje parcial</b>				<b>36</b>	<b>30</b>
<b>Puntaje total</b>	<b>66</b>				

Nota: Índice de validación del juicio de experto (Ivje) = [puntaje obtenido / 75] x 100=.....88.

#### 4. Escala de validación

Muy baja	Baja	Regular	Alta	Muy Alta
00-20 %	21-40 %	41-60 %	61-80%	81-100%
El instrumento de investigación está observado			El instrumento de investigación requiere reajustes para su aplicación	El instrumento de investigación está apto para su aplicación
<b>Interpretación:</b> Cuanto más se acerque el coeficiente a cero (0), mayor error habrá en la validez				

**5. Conclusión general de la validación y sugerencias** (en coherencia con el nivel de validación alcanzado):...El instrumento de recojo de información guarda coherencia con las variables de la investigación y por lo tanto es un instrumento válido para su aplicación.....

.....

### 6. Constancia de Juicio de experto

El que suscribe, ...EDGAR BAJALQUI CARRASCO.... identificado con DNI. N.º .....46129554.....

certifico que realicé el juicio del experto al instrumento diseñado por el (los) testistas

1...ROBERTO CARLOS DE LA CRUZ CERON.....

, en la investigación denominada: ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS DE  
DESARROLLO PARA APLICACIONES MÓVILES HÍBRIDAS.....

  
.....  
Firma del experto

### Anexos

Nº 1: Instrumento de investigación

Nº 2: Categorías investigativas

- Título de la investigación
- Formulación del problema
- Objetivo general
- Objetivos específicos
- Hipótesis (opcional en las investigaciones básicas)
- Operacionalización de variables

Nº 3: Evidencia de la prueba piloto (al menos un modelo)

## GUÍA, JUICIO DE EXPERTOS

### 1. Identificación del Experto

Nombre y Apellidos: ...JUAN ERICK LÓPEZ DE LA CRUZ.....

Centro laboral: FUNDACIÓN FONDO DE GARANTÍA PARA PRÉSTAMOS A LA PEQUEÑA INDUSTRIA - FOGAPI.....

Título profesional: .....INGENIERO DE SISTEMAS.....

Grado: ..... Mención.....

Institución donde lo obtuvo: UNIVERSIDAD CÉSAR VALLEJO.....

Otros estudios.....

### 2. Instrucciones

Estimado(a) especialista a continuación se muestra un conjunto de indicadores, el cual tienes que evaluar con criterio ético y estrictez científica, la validez del instrumento propuesto (véase anexo N° 1). Para evaluar dicho instrumento, marca con un aspa(x) una de las categorías contempladas en el cuadro:

1: Inferior al básico 2: Básico 3: Intermedio 4: Sobresaliente 5: Muy sobresaliente

### 3. Juicio de experto

INDICADORES	CATEGORÍA				
	1	2	3	4	5
1. Las dimensiones de la variable responden a un contexto teórico de forma (visión general)				X	
2. Coherencia entre dimensión e indicadores (visión general)				X	
3. El número de indicadores, evalúan las dimensiones y por consiguiente la variable seleccionada (visión general)				X	
4. Los ítems están redactados en forma clara y precisa, sin ambigüedades (claridad y precisión)					X
5. Los ítems guardan relación con los indicadores de las variables(coherencia)					X
6. Los ítems han sido redactados teniendo en cuenta la prueba piloto (pertinencia y eficacia)			X		
7. Los ítems han sido redactados teniendo en cuenta la validez de				X	

contenido					
8. Presenta algunas preguntas distractoras para controlar la contaminación de las respuestas (control de sesgo)				X	
9. Los ítems han sido redactados de lo general a lo particular(orden)				X	
10. Los ítems del instrumento, son coherentes en términos de cantidad(extensión)				X	
11. Los ítems no constituyen riesgo para el encuestado(inocuidad)					X
12. Calidad en la redacción de los ítems (visión general)				X	
13. Grado de objetividad del instrumento (visión general)					X
14. Grado de relevancia del instrumento (visión general)				X	
15. Estructura técnica básica del instrumento (organización)				X	
<b>Puntaje parcial</b>			3	40	20
<b>Puntaje total</b>	<b>63</b>				

Nota: Índice de validación del juicio de experto (Ivje) = [puntaje obtenido / 75] x 100=.....84.

#### 4. Escala de validación

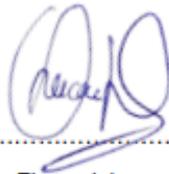
Muy baja	Baja	Regular	Alta	Muy Alta
00-20 %	21-40 %	41-60 %	61-80%	81-100%
El instrumento de investigación está observado			El instrumento de investigación requiere reajustes para su aplicación	El instrumento de investigación está apto para su aplicación
<b>Interpretación:</b> Cuanto más se acerque el coeficiente a cero (0), mayor error habrá en la validez				

**5. Conclusión general de la validación y sugerencias** (en coherencia con el nivel de validación alcanzado):.....El presente instrumento cumple con los indicadores establecidos de manera que es un instrumento apto para su aplicación en la

investigación.....  
.....

#### **6. Constancia de Juicio de experto**

El que suscribe, ...JUAN ERICK LÓPEZ DE LA CRUZ... identificado con DNI. N.º..... 45989478 ....  
certifico que realicé el juicio del experto al instrumento diseñado por el (los) tesisistas  
1...ROBERTO CARLOS DE LA CRUZ CERON.....  
en la investigación denominada: ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS DE  
DESARROLLO PARA APLICACIONES MÓVILES HÍBRIDAS.....



.....  
Firma del experto

#### **Anexos**

Nº 1: Instrumento de investigación

Nº 2: Categorías investigativas

- Título de la investigación
- Formulación del problema
- Objetivo general
- Objetivos específicos
- Hipótesis (opcional en las investigaciones básicas)
- Operacionalización de variables

Nº 3: Evidencia de la prueba piloto (al menos un modelo)

## GUÍA, JUICIO DE EXPERTOS

### 1. Identificación del Experto

Nombre y Apellidos: ...ANA MARÍA GUERRERO MILLONES.....

Centro laboral: .....UNIVERSIDAD SEÑOR DE SIPAN.....

Título profesional: .....INGENIERO .....

Grado: .....MAESTRA EN CIENCIAS. Mención: MAESTRA EN CIENCIAS.....

Institución donde lo obtuvo: UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO.....

Otros estudios.....LICENCIADA EN EDUCACIÓN, BACHILLER EN ADMINISTRACIÓN PÚBLICA.....

### 2. Instrucciones

Estimado(a) especialista a continuación se muestra un conjunto de indicadores, el cual tienes que evaluar con criterio ético y estrictez científica, la validez del instrumento propuesto (véase anexo N° 1).

Para evaluar dicho instrumento, marca con un aspa(x) una de las categorías contempladas en el cuadro:

1: Inferior al básico 2: Básico 3: Intermedio 4: Sobresaliente 5: Muy sobresaliente

### 3. Juicio de experto

INDICADORES	CATEGORÍA				
	1	2	3	4	5
1. Las dimensiones de la variable responden a un contexto teórico de forma (visión general)				X	
2. Coherencia entre dimensión e indicadores (visión general)				X	
3. El número de indicadores, evalúan las dimensiones y por consiguiente la variable seleccionada (visión general)					X
4. Los ítems están redactados en forma clara y precisa, sin ambigüedades (claridad y precisión)					X
5. Los ítems guardan relación con los indicadores de las variables(coherencia)				X	
6. Los ítems han sido redactados teniendo en cuenta la prueba piloto (pertinencia y eficacia)				X	
7. Los ítems han sido redactados teniendo en cuenta la validez de				X	

contenido					
8. Presenta algunas preguntas distractoras para controlar la contaminación de las respuestas (control de sesgo)			X		
9. Los ítems han sido redactados de lo general a lo particular(orden)					X
10. Los ítems del instrumento, son coherentes en términos de cantidad(extensión)					X
11. Los ítems no constituyen riesgo para el encuestado(inocuidad)					X
12. Calidad en la redacción de los ítems (visión general)				X	
13. Grado de objetividad del instrumento (visión general)				X	
14. Grado de relevancia del instrumento (visión general)				X	
15. Estructura técnica básica del instrumento (organización)				X	
<b>Puntaje parcial</b>			<b>3</b>	<b>36</b>	<b>25</b>
<b>Puntaje total</b>	<b>64</b>				

Nota: Índice de validación del juicio de experto (IVJE) = [puntaje obtenido / 75] x 100=.....85.3.

#### 4. Escala de validación

Muy baja	Baja	Regular	Alta	Muy Alta
00-20 %	21-40 %	41-60 %	61-80%	81-100%
El instrumento de investigación está observado			El instrumento de investigación requiere reajustes para su aplicación	El instrumento de investigación está apto para su aplicación
<b>Interpretación:</b> Cuanto más se acerque el coeficiente a cero (0), mayor error habrá en la validez				

**5. Conclusión general de la validación y sugerencias** (en coherencia con el nivel de validación alcanzado): Se puede evidenciar que existe coherencia entre las variables, dimensiones e indicadores los cuales responden a los objetivos planteados en la investigación por lo tanto, el instrumento está

apto para su aplicación.....  
.....

#### **6. Constancia de Juicio de experto**

El que suscribe, ...ANA MARÍA GUERRERO MILLONES... identificado con DNI. N.º 17535600  
certifico que realicé el juicio del experto al instrumento diseñado por el (los) tesisistas

1...ROBERTO CARLOS DE LA CRUZ CERON.....

en la investigación denominada: ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS DE  
DESARROLLO PARA APLICACIONES MÓVILES HÍBRIDAS.....



.....  
Firma del experto

#### **Anexos**

Nº 1: Instrumento de investigación

Nº 2: Categorías investigativas

- Título de la investigación
- Formulación del problema
- Objetivo general
- Objetivos específicos
- Hipótesis (opcional en las investigaciones básicas)
- Operacionalización de variables

Nº 3: Evidencia de la prueba piloto (al menos un modelo)

Anexo 3. Matriz de Consistencia Lógica de tema de investigación

<b>MATRÍZ DE CONSISTENCIA LÓGICA DE TEMA DE INVESTIGACIÓN</b>				
<b>Enfoque metodológico</b>				
<b>Título</b>	<b>ANÁLISIS COMPARATIVO ENTRE FRAMEWORKS DE DESARROLLO PARA APLICACIONES MÓVILES HÍBRIDAS</b>			
<b>Apellidos y nombres (1)</b>	<b>DE LA CRUZ CERON ROBERTO CARLOS</b>			
	<b>Problema</b>	<b>Variables</b>		
	¿Cómo elegir el framework que mejor se adapte a las necesidades de un equipo de desarrollo a la hora de implementar una aplicación móvil híbrida?	- Frameworks - Aplicaciones móviles híbridas		
	<b>Hipótesis</b>	<b>Objetivo General</b>	<b>Objetivos específicos</b>	<b>Método propuesto</b>
	Se implementa una aplicación móvil híbrida con CRUD con un cada uno de los frameworks seleccionados para medir los resultados mediante casos de pruebas.	Realizar análisis comparativo de los frameworks para el desarrollo de aplicaciones móviles híbridas.	a. Seleccionar los frameworks para realizar la comparativa. b. Desarrollar una aplicación como caso de estudio para la comparación. c) Realizar pruebas a las aplicaciones desarrolladas con base en los frameworks.	a. Obj.A 1.- Buscar frameworks de desarrollo para aplicaciones móviles híbridas. 2.- Revisar documentación y tutoriales sobre cada uno. 3.- Seleccionar dos frameworks a analizar. 4.- Validar documentación, instaladores y tutoriales. b. Obj.b 1.- Definir el negocio de la aplicación a desarrollar. 2.- Iniciar desarrollo de la aplicación con cada framework. 3.- Realizar entrevista a equipo de desarrollo. c. Objc 1.- Presentar y analizar resultados de encuestas. 2.- Revisar y analizar métricas obtenidas en el desarrollo de la aplicación. 3.- Contrastar con la hipótesis.

