



**FACULTAD DE INGENIERIA, ARQUITECTURA Y  
URBANISMO**

**ESCUELA ACADÉMICO PROFESIONAL DE  
INGENIERÍA DE SISTEMAS**

**TESIS  
EVALUACIÓN COMPARATIVA DE  
FRAMEWORKS DE DESARROLLO PARA MEDIR  
EL RENDIMIENTO EN APLICACIONES MÓVILES  
ANDROID**

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

**Autor:**

**Bach. Guevara Zavaleta Noheli Elizabeth**

**Asesor:**

**Mg. Tuesta Monteza Víctor Alexci**

**Línea de Investigación:  
Tecnologías de la Información**

**Pimentel – Perú  
2017**



EVALUACIÓN COMPARATIVA DE FRAMEWORKS DE  
DESARROLLO PARA MEDIR EL RENDIMIENTO EN  
APLICACIONES MÓVILES ANDROID

Aprobación de la tesis

---

Ing. Mejía Cabrera Heber Iván  
**Presidente del jurado de tesis**

---

Ing. Samillan Ayala Alberto Enrique  
**Secretario del jurado de tesis**

---

Ing. Ocampo Moreno Rosa Liliana  
**Vocal del jurado de tesis**

## DEDICATORIA

A Dios por ser mí guía, en cada paso que doy día a día guiarme en los Momentos difíciles de mi vida.

A mis queridos Padres y hermanas, que con su esfuerzo y apoyo incondicional me permitieron salir adelante, con empeño optimismo, representando mi fortaleza, esfuerzo y dedicación de seguir superándome.

A mi abuelita, que ya no está a lado mío, pero su cariño prevalece siempre en mi corazón.

A mis amigos con quienes he tenido la oportunidad de compartir mis conocimientos y me han ayudado a reforzar y adquirir nuevas experiencias.

**Noheli Guevara**



## AGRADECIMIENTO

A toda mi familia y compañeros de estudios por su apoyo constante de entrega y sacrificio, al tiempo de lucha constante y gratas vivencias, de momentos de éxitos y también de angustias para poder cumplir nuestros objetivos y así poder alcanzar uno de nuestros más grandes anhelos, culminar nuestra carrera, los deseos de superarnos y vencer todos los obstáculos y es por ello que dedicamos este triunfo a quienes en todo momento nos llenaron de amor y ayuda.

Los profesores en general por la labor académica desempeñada en la formación profesional de nuestra persona.

**Noheli Guevara**



## INDICE

<b>CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN</b> .....	12
1.1. Situación problemática.....	12
1.2. Formulación del problema.....	13
1.3. Delimitación de la Investigación.....	13
1.4. Justificación e importancia de la investigación.....	13
1.5. Limitaciones de la investigación.....	14
1.6. Objetivos de la investigación.....	14
<b>CAPÍTULO II: MARCO TEÓRICO</b> .....	16
2.1. Antecedentes de la investigación.....	16
2.2. Estado del arte.....	20
2.3. Base teórica científicas.....	24
2.3.1. Aplicaciones móviles.....	24
2.3.2. Entornos de desarrollo.....	28
2.3.3. Plataformas móviles.....	28
2.3.4. Librerías y Frameworks Móviles.....	35
2.3.5. Calidad de Software.....	35
2.3.6. Factores de Calidad de Software.....	36
2.3.7. Métricas de Calidad.....	37
2.3.8. Norma ISO/IEC 25000.....	37
2.3.9. ISO/IEC 25010.....	37
2.4. Definición de términos básicos.....	39
<b>CAPÍTULO III: MARCO METODOLÓGICO</b> .....	40
3.1. Tipo y diseño de investigación.....	40
3.1.1. Tipo de investigación.....	40
3.1.2. Diseño de la investigación.....	40
3.2. Población y muestra.....	40
3.2.1. Población.....	40
3.2.2. Muestra.....	40
3.3. Hipótesis.....	40
El framework Apache Cordova es el más óptimo en cuanto a rendimiento. ....	40
3.4. Variables.....	40
3.4.1. Variable dependiente.....	40
3.4.2. Variable independiente.....	40



3.5.	Operacionalización .....	41
3.6.	Abordaje metodológico, técnicas e instrumentos de recolección de datos	42
3.6.1.	Abordaje metodológico .....	42
3.6.2.	Técnicas de recolección de datos .....	42
3.6.3.	Instrumentos de recolección de datos .....	42
3.7.	Procedimiento para la recolección de datos .....	43
3.8.	Análisis estadístico e interpretación de los datos .....	43
3.9.	Principios éticos .....	44
3.10.	Criterios de rigor científico .....	44
<b>CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS .....</b>		<b>45</b>
4.1.	Resultados en tablas y gráficos .....	45
4.2.	Discusión de resultados .....	49
<b>CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN .....</b>		<b>50</b>
5.1	Seleccionar los frameworks para el desarrollo de aplicaciones móviles...	51
5.2	Seleccionar los parámetros con el cual se evaluará la aplicación móvil.	56
5.2.1	Definir el proceso de evaluación.....	57
5.2.2	Selección de subcaracterísticas y atributos de calidad para APPEAT.....	58
5.3	Desarrollar una aplicación móvil haciendo uso de los frameworks seleccionados .....	63
5.3.1	A Domicilio Perú SAC .....	63
5.3.2	Desarrollo de la aplicación móvil ADP EAT .....	64
5.3.3	Planificación para el desarrollo de la aplicación móvil.....	65
5.3.4	Historias de usuario .....	68
5.3.5	Product Backlog .....	69
5.3.6	Sprint Planning Meeting .....	70
5.3.7	SPRINT 01.....	71
5.4	Comparar las aplicaciones desarrolladas con los frameworks.....	82
5.4.1	Medir la calidad externa de las aplicaciones móviles .....	82
6.1.	Conclusiones.....	106
6.2.	Recomendaciones.....	107
<b>REFERENCIAS .....</b>		<b>108</b>



## Índice de ilustraciones

<b>Ilustración 1</b> Ciclo de vida de una Actividad Android .....	32
<b>Ilustración 2</b> Características ISO 25000 .....	38
<b>Ilustración 3</b> resultado eficiencia en el desempeño .....	45
<b>Ilustración 4</b> Resultados de tiempo de respuesta .....	46
<b>Ilustración 5</b> Resultados tiempo de espera .....	46
<b>Ilustración 6</b> Resultado de rendimiento.....	47
<b>Ilustración 7</b> Resultados de utilización de CPU .....	48
<b>Ilustración 8</b> Resultado de utilización de memoria RAM.....	48
<b>Ilustración 9</b> Diagrama de flujo que muestra el proceso de la propuesta planteada...	50
<b>Ilustración 10</b> Cuadrante Mágico .....	51
<b>Ilustración 11</b> Organigrama de A domicilio Perú SAC.....	63
<b>Ilustración 12</b> Metodología SRUM .....	64
<b>Ilustración 13</b> arquitectura del aplicativo.....	72
<b>Ilustración 14</b> diseño de base de datos .....	73
<b>Ilustración 15</b> base de datos MySQL .....	75
<b>Ilustración 16</b> servidor de archivos .....	76
<b>Ilustración 17</b> Conexión BD .....	76
<b>Ilustración 18</b> clase métodos.php .....	77
<b>Ilustración 19</b> librerías de JQuery Mobile .....	78
<b>Ilustración 20</b> Versión de SO Android.....	79

## Índice de tablas

<b>Tabla 1</b> evaluación de frameworks.....	52
<b>Tabla 2</b> evaluación multiplataforma .....	53
<b>Tabla 3</b> evaluación de edad del software.....	53
<b>Tabla 4</b> evaluación de costo del software .....	54
<b>Tabla 5</b> evaluación Open Source .....	55
<b>Tabla 6</b> evaluación de documentación .....	55
<b>Tabla 7</b> evaluación de popularidad .....	56
<b>Tabla 8</b> descripción de tareas .....	57
<b>Tabla 9</b> Nivel de importancia.....	58
<b>Tabla 10</b> subcaracterísticas y atributos de calidad externa seleccionadas .....	58
<b>Tabla 11</b> métricas para calidad externa .....	59
<b>Tabla 12</b> Matriz de calidad externa .....	60
<b>Tabla 13</b> Criterios de la matriz de calidad.....	61
<b>Tabla 14</b> Matriz de Calidad Externa.....	62
<b>Tabla 15:</b> Roles de SCRUM.....	65
<b>Tabla 16</b> formato de historia de usuario .....	68
<b>Tabla 17</b> Formato de historia técnica.....	68
<b>Tabla 18</b> Product Backlog.....	69
<b>Tabla 19</b> Planificación de la aplicación móvil .....	69
<b>Tabla 20</b> Sprint BackLog - Sprint 1 .....	70
<b>Tabla 21</b> Sprint Backlog - Sprint 2.....	71



<b>Tabla 22</b> Sprint Backlog - Sprint 3.....	71
<b>Tabla 23</b> Diccionario de datos.....	75
<b>Tabla 24</b> Tiempo de Respuesta.....	84
<b>Tabla 25</b> Tiempo de respuesta – sumatoria .....	85
<b>Tabla 26</b> Tiempo de espera.....	86
<b>Tabla 27</b> Tiempo de espera – sumatoria.....	87
<b>Tabla 28</b> Tiempo de Rendimiento .....	88
<b>Tabla 29</b> Tiempo de rendimiento – sumatoria .....	89
<b>Tabla 30</b> Utilización de CPU .....	90
<b>Tabla 31</b> Utilización de CPU – Sumatoria.....	91
<b>Tabla 32</b> Utilización de la memoria .....	92
<b>Tabla 33</b> Utilización de memoria – sumatoria .....	93
<b>Tabla 34</b> Matriz de calidad externa .....	94
<b>Tabla 35</b> Tiempo de respuesta .....	95
<b>Tabla 36</b> Tiempo de respuesta – sumatoria .....	96
<b>Tabla 37</b> Tiempo de espera.....	97
<b>Tabla 38</b> Tiempo de espera – sumatoria.....	98
<b>Tabla 39</b> Tiempo de Rendimiento .....	99
<b>Tabla 40</b> Tiempo de rendimiento – sumatoria .....	100
<b>Tabla 41</b> Utilización de CPU .....	101
<b>Tabla 42</b> Utilización de CPU – sumatoria .....	102
<b>Tabla 43</b> Utilización de memoria .....	103
<b>Tabla 44</b> Utilización de memoria – sumatoria .....	104
<b>Tabla 45</b> Matriz de calidad externa .....	105





## RESUMEN

Esta investigación surge debido a que las aplicaciones móviles son cada vez más importantes en la sociedad actual. Se estima que un gran porcentaje de usuarios no están satisfechos, por el consumo excesivo de memoria, batería, CPU, y características relacionadas a la eficiencia, siendo un factor de calidad de producto software.

Por ende, el presente trabajo de investigación plantea como objetivo principal comparar frameworks de desarrollo para medir el rendimiento en aplicaciones móviles, siguiendo el estándar ISO 25000 para evaluación de cada aplicación, desarrollada bajo la metodología ágil; mediante la herramienta de evaluación de rendimiento, que proporciona Android Studio. La selección de los frameworks se realizó de acuerdo a un top realizado en el presente año por (G2 Crowd, 2017), realizando un cuadro de comparación, sobre las siguientes características: edad del software, desarrollo multiplataforma, costo del software, open source, documentación y popularidad, siendo elegidos los frameworks apache cordova y JQuery Mobile. El desarrollo de las aplicaciones consiste en un CRUD (crea, editar, actualizar y eliminar), siendo una arquitectura de cliente servidor, haciendo peticiones a un web service externo alojado en un hosting.

El rendimiento de las aplicaciones móviles se realizó con los siguientes indicadores para comportamiento en el tiempo (tiempo de respuesta, tiempo de espera, rendimiento) y utilización de recursos (uso de CPU, uso de memoria RAM). Utilizando como software de testeo Android Studio 3.0 canary, ofreciendo un conjunto de herramientas para medir y diagnosticar rápidamente problemas de rendimiento de las aplicaciones. Se ha considerado realizar 30 pruebas de testeo para cada indicador, obteniendo un promedio, para ser procesado con su respectiva formula según Norma ISO 25010.

Como resultado se obtuvo que la aplicación móvil desarrollada con el framework JQuery Mobile tiene un mejor rendimiento, menor uso de recursos de memoria y CPU, en comparación a la aplicación desarrollada con framework apache cordova.



**Palabras claves:** framework, ISO 25000, rendimiento, memoria RAM, CPU.

## ABSTRACT

This research arises because mobile applications are increasingly important in today's society.

It is estimated that a large percentage of users are not satisfied, due to the excessive consumption of memory, battery, CPU, and features related to efficiency, being a factor of software product quality

Therefore, the main objective of this research is to compare development frameworks to measure performance in mobile applications, following the ISO 25000 standard for evaluation of each application, developed under the agile methodology; Using the performance evaluation tool provided by Android Studio.

The selection of the frameworks was done according to a top realized in the present year by (G2 Crowd, 2017), making a comparison table, on the following characteristics: software age, multiplatform development, software cost, open source, Documentation and popularity, being chosen the frameworks apache cordova and JQuery Mobile.

As a result it was obtained that the application developed with the framework apache cordova has a better performance, less use of memory resources and CPU, compared to the application developed with framework JQuery Mobile.

**Keywords:** Framework, ISO 25000, performance, RAM, CPU.



## INTRODUCCIÓN

Las aplicaciones móviles se han convertido en parte fundamental de la vida cotidiana desde que surgieron hace alrededor de un cuarto de siglo, con el aprovechamiento de los avances tecnológicos, el desarrollo de aplicaciones está a la vanguardia y ha provocado un sinnúmero de mejoras en el desempeño del ser humano (Careaga, 2017), Sin embargo cada vez los usuarios son más exigentes en la calidad de las aplicaciones móviles, por ello en diferentes lugares del mundo se investiga sobre cómo mejorar la calidad de producto, esto debido a que los desarrolladores de software han tomado de manera ligera el tema de calidad del producto, haciendo escaso uso de los estándares Internacionales tales como ISO/IEEE que miden temas de eficiencia, usabilidad, funcionalidad, mantenibilidad. (Sanjuán, Rodríguez, & Rosado, 2015)

Según la ISO/IEC 25010, la eficiencia representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Es por ello que en esta investigación se pretende atender un atributo de calidad que se presenta en la mayoría de clasificaciones, que es la eficiencia de desempeño. La cual especifica el grado en que los usuarios pueden utilizar un producto de forma gratificante, logrando alcanzar aspectos como efectividad y eficiencia en el contexto de uso. Debido a que la eficiencia es un aspecto crítico en la interactividad de aplicaciones móviles, por ello es un factor determinante en la aceptación de los productos software, implicando el éxito del mismo.



## CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN

### 1.1. Situación problemática

En los últimos años las aplicaciones móviles están presentes en la mayoría de la población, y han permitido a numerosos usuarios de dispositivos móviles acceder a los recursos web con los teléfonos inteligentes (Aguado, Martínez, & Cañete-Sanz, 2015), quien también menciona que el uso de las aplicaciones móviles ha ido en aumento, porque resultan ser muy atractivos para los usuarios, debido a las interfaces gráficas de fácil uso y a los servicios especializados que ofrece: como aplicaciones de personalización, productividad, estilo de vida, compras entre otros, que mediante la explotación de las capacidades avanzadas de los teléfonos inteligentes, las aplicaciones móviles se pueden adaptar para satisfacer las necesidades específicas de los usuarios.

Sin embargo cada vez los usuarios son más exigentes en la calidad de las aplicaciones móviles, por ello en distintas partes del mundo se investiga sobre cómo mejorar la calidad de producto, esto debido a que los desarrolladores de software han tomado de manera ligera el tema de calidad del producto, haciendo escaso uso de los estándares Internacionales tales como ISO/IEEE que miden temas de eficiencia, usabilidad, funcionalidad, mantenibilidad.(Sanjuán et al., 2015), afirma que: “las características de calidad [de producto] indispensables en las aplicaciones son: funcionalidad, seguridad, usabilidad y rendimiento, en el contexto de las aplicaciones móviles la eficiencia o también llamada rendimiento, se destaca notablemente [...] En este sentido el rendimiento en las aplicaciones es un factor importante ya que depende de ello, que el usuario adquiera una experiencia única y positiva al hacer uso de ellas. Según (Liu, Xu, & Cheung, 2015), menciona que diversas aplicaciones tienen errores: como pérdida de energía de la batería, el rendimiento puede reducir significativamente la respuesta de los smartphones y los recursos computacionales disponibles como la memoria y ancho de



banda de red, de manera que estos errores afectan gravemente a la experiencia del usuario y causa una significativa frustración del mismo.

Por lo tanto existen herramientas que facilitan el desarrollo de aplicaciones móviles, y están cobrando alta importancia debido a la serie de plataformas móviles existentes, de las cuales, Android es la más sobresaliente, por la seguridad que ofrece; por ello los analistas creen que Android será líder mundial [en la actualidad ya lo es], por lo tanto el futuro de Android es prometedor, ya que desde el punto de vista arquitectónico, las aplicaciones de Android dan mayor seguridad que las aplicaciones de iPhone. (Butler, 2011)

En consecuencia, el uso de las aplicaciones móviles es muy frecuente, sin embargo, hoy en día existen debilidades en la calidad del producto, teniendo como resultado una mala experiencia del usuario. Por ello se viene realizando diferentes estudios para contribuir a la mejora del desarrollo de las aplicaciones móviles teniendo en cuenta las normas de Calidad del producto, cuya característica de mayor importancia es el rendimiento.

## **1.2. Formulación del problema**

¿Cuál es el framework de desarrollo en aplicaciones móviles Android con mejor rendimiento?

## **1.3. Delimitación de la Investigación**

La presente investigación se enfocará en evaluar frameworks de desarrollo mediante una aplicación móvil.

La cantidad de frameworks de desarrollo a evaluar son 2, los cuales son de transformación móvil.

La evaluación se realizará en torno al atributo de calidad rendimiento, teniendo como indicadores para comportamiento en el tiempo (tiempo de respuesta, tiempo de espera, rendimiento) y utilización de recursos (uso de CPU, uso de memoria RAM).

## **1.4. Justificación e importancia de la investigación**

La investigación estuvo orientada en estudiar los frameworks de desarrollo móvil, desde la perspectiva de rendimiento, tuvo en cuenta el



código fuente para la optimización de uso de recursos del teléfono inteligente, ya que, debido al creciente uso de las aplicaciones móviles, presentaron una fuerte debilidad en calidad de rendimiento. Así mismo contribuyó en la solución del problema del rendimiento que afecta a un gran porcentaje de aplicaciones móviles.

La investigación se justificó porque en la medida que se desarrolló las aplicaciones móviles con frameworks y estas cumplieron con los estándares de calidad en rendimiento y optimización de hardware. Así mismo permitió determinar que frameworks cumplen con los estándares de calidad de software, así mismo la optimización de software y hardware. También ha permitido generar nuevos espacios de conocimiento, los cuales deben ser aprovechados, para proponer uso de tecnologías más eficientes, puesto que los problemas de rendimiento y alto consumo de energía por aplicaciones móviles en los Smartphone conllevan a altos costos económicos, por tanto, lograr resolver dichos problemas generan ganancias, tanto para el usuario y también a desarrolladores.

Finalmente realizar este estudio permitió analizar y seleccionar los frameworks que permitieron mejorar el rendimiento y la optimización de recursos del equipo móvil, logrando obtener un producto de calidad de acuerdo con la norma ISO 25010, lo cual significó un producto de éxito. De esta manera se generó nuevos conocimientos, dando un aporte valioso al desarrollo de esta línea de investigación.

## **1.5. Limitaciones de la investigación**

Limitada información de la norma ISO 25000.

## **1.6. Objetivos de la investigación**

### **Objetivo general**

Comparar frameworks de desarrollo para medir el rendimiento en aplicaciones móviles.



## Objetivos específicos

1. Seleccionar los frameworks para el desarrollo de aplicaciones móviles.
2. Seleccionar los parámetros con el cual se evaluará la aplicación móvil.
3. Desarrollar una aplicación móvil haciendo uso de los frameworks seleccionados.
4. Comparar los frameworks en las aplicaciones desarrolladas.



## CAPÍTULO II: MARCO TEÓRICO

### 2.1. Antecedentes de la investigación

(Rengifo & Betancourt, 2011) "*Frameworks y Herramientas para el Desarrollo de Aplicaciones orientadas a dispositivos móviles*" (Tesis de Grado). Universidad tecnológica de Pereira, Colombia. Tiene como objetivo desarrollar un documento monográfico de las herramientas y frameworks para el desarrollo de aplicaciones en dispositivos móviles. Como primera etapa inician una recopilación del estado del arte, utilizando herramientas como internet, papers, artículos, noticias, libros entre otros, sobre varias herramientas y frameworks para el desarrollo de aplicaciones orientadas a dispositivos móviles. Como segunda etapa realizaron una segmentación de la recopilación realizada de acuerdo al público dirigido, como es el mercado masivo, puntual o nichos de mercado. Y como etapa final realizaron un análisis de las herramientas y frameworks para comparar sus ventajas y desventajas en términos de eficiencia, portabilidad, escalabilidad, entre otros, a la hora de desarrollar cualquier tipo de aplicación. Teniendo la siguiente conclusión, la elección de la herramienta o framework a utilizar, solo depende de las necesidades que la aplicación debe cumplir al tener en cuenta aspectos como tipo de aplicación a desarrollar, los dispositivos en los cuales se instalará, plataforma de software utilizada e IDE entre otros.

En Agosto de 2014, (Andrés, 2014), realizó un estudio sobre "*Estudio comparativo de diferentes Frameworks de alto nivel para el desarrollo de aplicaciones Móviles en Android*" (Proyecto Final de Master).

Otro tipo de empresas de desarrollo de aplicaciones como también otros desarrolladores autónomos comparten el mismo problema y es la variedad de sistemas que existen en el ámbito móvil pues si se escoge uno se deja afuera a otros potenciales clientes o negocios. Debido a esto optan por escoger este tipo de frameworks el cual brindan una solución más acertada con respecto a tener que gastar tiempo y dinero en aprender otra tecnología. En este estudio se centró más en buscar cómo crear





aplicaciones móviles en Android con menor esfuerzo, tiempo y que se pueda implementar mediante tecnologías web estándar u otras tecnologías mucho más flexibles. Por tanto, se compararon una con otra de acuerdo a un conjunto de componentes principales de Android tal como Actividades, servicios, Broadcast Receivers, IU, Persistencia, animación, multimedia, sensores y autolayout, por último, se comparan de acuerdo a las redes sociales soportadas por cada una. Teniendo como conclusión que Corona es uno de los frameworks más fuertes en la parte visual y de animación en el desarrollo de una aplicación móvil para Android, Titanium brinda la mejor experiencia nativa de todos los frameworks, no solo por su API sino porque se centra más en la simplicidad en el desarrollo de una aplicación móvil con tecnologías estándar web, Parse es el más indicado para permitir el acceso mediante redes sociales como Facebook o Twitter además que también tienen compatibilidad con otros frameworks como es el caso de Titanium.

En noviembre de 2014 se realizó un estudio por (Bender & Chavez, 2014) *“Desarrollo de aplicaciones híbridas multiplataforma con PhoneGap para dispositivos HandHeld”*. En este trabajo de investigación realizaron un análisis del framework PhoneGap, este análisis comprendió los aspectos teóricos de mayor relevancia, así como también realización de una prueba empírica para el desarrollo de la herramienta seleccionada, teniendo en cuenta los tres paradigmas de programación existentes como la aplicaciones móviles nativas, web e híbrido, la investigación se centró en aplicaciones móviles híbridas motivo por el cual implicó la investigación de lenguajes y tecnologías relacionadas con el framework, así también como el uso de buenas prácticas y librerías. Se tuvo en cuenta que, para el desarrollo en dispositivos móviles, se analizaron los que tienen mayor presencia en el mercado, observando sus características, ventajas, compatibilidad y versiones de donde se planteó el por qué se eligió el framework PhoneGap para su experimentación. Para poder tener un escenario donde poner a prueba el framework seleccionado se desarrolló



la aplicación móvil “DigitalHC”, la cual gestiona una pequeña historia clínica de pacientes, la cual se ejecutó sobre dispositivos con plataforma Android con dichas pruebas se pretendió determinar las capacidades reales de PhoneGap sobre la plataforma seleccionada. Para finalizar luego del análisis, investigación, desarrollo, evolución y pruebas realizadas, los estudios a los que concluyeron con este trabajo se plasmaron en resultados en cual llegaron a la conclusión de que Phonegap es una herramienta que nos permite mejorar la capacidad de desarrollo y en cuanto costo es relativamente bajos en comparación con las aplicaciones móviles nativas. Sera un referente para otra investigación futuras y para los programadores en el momento que tenga que decidir que herramienta multiplataforma utilizar de las que se encuentran en mercado.

En el año 2014 se realizó un estudio por (Hidalgo Orozco & Casigña Parra, 2015) “*Análisis del Rendimiento entre IOS y Android para Desarrollar el Sistema de Posicionamiento y Administración para Taxis Ejecutivos*”. Escuela Superior Politécnica De Chimborazo, Ecuador. En este trabajo de investigación analizaron el rendimiento de los teléfonos móviles entre iPhone con iOS y Samsung con Android para desarrollar el Sistema de Posicionamiento y Administración para Taxis Ejecutivos. Para realizar la selección de la plataforma de mejor rendimiento realizaron un análisis en base al consumo de recursos, estableciendo como indicadores el uso de RAM y CPU, en el análisis que realizaron aplicaron modelos de estadística descriptiva, que evidencia la diferencia entre las muestras de datos analizados, teniendo como resultado el dispositivo que menor consumo de recursos utiliza al ejecutar la aplicación “Taxímetro”. En conclusión, de la comparación realizada en la presente investigación, determinaron que los teléfonos móviles iPhone con iOS, son mejor en cuanto a rendimiento con 51% y 3% menor costo que los dispositivos Samsung con Android.



En el mismo año 2014, (Pérez Reyes, 2014), hizo un estudio sobre “*Análisis de rendimiento de aplicaciones Android*” (Trabajo de Fin de Grado). Universidad de la Laguna, España. En el cual tenía como objetivo el desarrollo, y análisis de rendimiento, de aplicaciones Android. Para lo cual plantearon como caso de estudio el algoritmo de procesamiento de imágenes conocido como la “Transformada de Hough”. Tras una fase de análisis se ha optado finalmente por adaptar a la plataforma Android un código base disponible para la detección de líneas en imágenes digitales. Esta adaptación implicaría, necesariamente, familiarizarse con el entorno Android siguiendo las pautas descritas en la página oficial de desarrollo de la plataforma. Para el análisis de rendimiento en Android, se siguieron las técnicas y herramientas descritas en la página oficial de desarrollo, que permitieron comprobar el tiempo de ejecución, las funciones computacionalmente más costosas y la memoria consumida. Una vez familiarizado con la detección de líneas, se decidió ampliar el estudio a la detección de círculos en imágenes digitales, adaptando y mejorando, igual que en el procedimiento anterior, el código base disponible para su uso en Android. Durante el estudio de las aplicaciones, se crearon nuevas versiones que mejoraban progresivamente los resultados obtenidos. Cuando se consiguió la versión más rápida usando código secuencial, se procedió a paralelizar la función o funciones computacionalmente más costosas usando Renderscript. Por último, se ha hecho uso del framework Paralldroid que permite paralelizar código Android automáticamente mediante anotaciones en el código Java. El proyecto finaliza con un análisis comparativo de los resultados obtenidos. A modo de conclusión, intentaron matizar aspectos que deben ser considerados al desarrollar aplicaciones en la plataforma Android que pueden tener impacto en el rendimiento.

En octubre de 2014 (Amparo, 2014), realizo un estudio sobre “*Evaluación de calidad de productos software en empresas de desarrollo de software aplicando la norma ISO/IEC 25000*”, Escuela Politécnica Nacional, Quito.



En el cual evaluó un sistema LogiNotificador, siguiendo el proceso de evaluación y personalizando el modelo de calidad que contiene la norma. Trataron sobre el análisis de las características y productos que ofrecen las empresas de desarrollo de software, luego realizaron una revisión del modelo de calidad de la ISO/IEC 25010 y especificaron el procedimiento con el cual se va a realizar la evaluación de acuerdo a la ISO/IEC 25040. Posteriormente realizaron la definición de características, subcaracterísticas y atributos de la calidad interna, externa y en uso con una descripción detallada de las métricas, para luego establecer el procedimiento de aplicación de la matriz de calidad a utilizarse en dicha evaluación. Se llevó a cabo un análisis del producto software a ser evaluado, especificando los requisitos del sistema siguiendo el estándar IEEE 830 y funcionalidad del mismo.

## 2.2. Estado del arte

(La & Kim, 2012) “Balanced MVC Architecture for High Efficiency Mobile Applications”. En esta investigación propone una arquitectura de MVC equilibrada para mejorar el rendimiento de las aplicaciones móviles en los teléfonos inteligentes, lo cual consiste en dejar que algunas funciones de gran peso ejecutando en el lado del servidor y dejar que una aplicación cliente acceda a la función en el servidor. Para llevar a cabo este tipo de aplicaciones móviles, adoptaron los principios de diseño de arquitectura bien definidos; siendo un cliente ligero, estar superpuestas con el Modelo-Vista-Controlador (MVC), y ser equilibrado entre cliente y servidor. Con la adopción de los principios, propusieron una arquitectura única, ideal y práctica para las aplicaciones móviles, llamada arquitectura MVC equilibrada. Por lo tanto, definieron criterios clave para determinar el grado de rendimiento, y un método para diseñar una arquitectura MVC equilibrada que incorpora funcionalidad de partición para el alto rendimiento, y un método de evaluación basado en la simulación de arquitecturas MVC equilibradas. Por lo tanto, el desarrollo de dicha investigación fue la siguiente: primero presentaron la arquitectura MVC



equilibrada y cinco patrones más específicos de la arquitectura, a continuación, se define criterios clave de determinación del rendimiento de las aplicaciones móviles con MVC equilibrada que pueden ser utilizados para el diseño y la evaluación de la arquitectura. Y presentaron un método sistemático para aplicar la arquitectura y un método de evaluación basado en la simulación para evaluar si el diseño de la arquitectura está completamente satisfecho con los criterios, y por último, presentaron tres casos de estudio que muestran cómo se aplica la arquitectura en la práctica, que muestra cómo la arquitectura es correctamente evaluada, y muestra resultados de comparación con otros patrones convencionales. Con las directrices de arquitectura y diseño propuesto, concluyeron que las aplicaciones móviles con alta complejidad se pueden diseñar con eficacia y con un alto rendimiento y bajo consumo de recursos.

(Raudjärv & Gregersen, 2015) *“JRebel.Android: recarga de recursos y clases de ejecución para Android”* Tartu, Estonia. En este artículo se presentó una herramienta para recargar clases y recursos en la plataforma Android. El cual tiene dos componentes, agente que se ejecuta en un dispositivo Android y el servicio que se ejecuta en el host (máquina del desarrollador). La responsabilidad principal del componente de servicio es estar atento a los cambios en las clases y recursos de directorios. Al encontrar cambios en los archivos del componente de servicio los prepara para recargas posteriores al transformar el código de bytes de acuerdo con el funcionamiento interno de JRebel y las envía al agente, contando el agente para llevar a cabo una tarea de recarga. Cada vez que el componente de agente recibe los archivos del componente de servicio, se dispara el tiempo de ejecución de la operación de recarga, por tanto, hacer los cambios sin conexión que fueron recogidos por el componente de servicio activo en el dispositivo. Esta herramienta se diseñó con la facilidad de uso, así que el único parámetro de entrada necesaria para el servicio es el directorio raíz de un proyecto Android. Teniendo en cuenta este directorio el componente de servicio localiza



automáticamente los directorios Android manifiesto de archivo, de clase y de recursos, así como ninguna biblioteca externa y dependencias del proyecto. Por lo general uno de los dos sistemas de construcción se utiliza en el desarrollo de aplicaciones para Android, que sean Ant o Gradle. JRebel.Android soporta ambos diseños, e incluso hay soporte para algunos diseños de proyectos a medida incorporados. Con esta herramienta, los desarrolladores que escriben aplicaciones pueden beneficiarse de un tiempo de construcción cercano a cero y de una retroalimentación inmediata de los cambios realizados, sin necesidad de pasar por un dificultoso escenario de reconstrucción y reinicio de toda la aplicación. Una de las principales áreas de enfoque ha sido la facilidad de uso, que se manifiesta de varias maneras, incluyendo los requisitos mínimos para la configuración. Se ha demostrado que funciona en dos aplicaciones de muestra, una de las cuales es bastante grande, y que la productividad obtenida al no tener que reiniciar es significativa.

(Syer, Nagappan, Adams, & Hassan, 2015) *“Studying the relationship between source code quality and mobile platform dependence”*. En este trabajo se presentó un estudio de la relación entre la plataforma dependencia y la propensión de defecto de los archivos de código fuente de una aplicación para Android. El estudio que realizaron para determinar si las métricas de software basados en la dependencia de la plataforma se pueden utilizar para priorizar los esfuerzos de aseguramiento de la calidad del software. Realizaron un estudio: Si los archivos de código fuente de defectos propensos son más dependientes de la plataforma Android de archivos de código fuente sin defectos. Si la tasa de dependencia de la plataforma puede ayudar a explicar estadísticamente explican defectos. Las fuentes métricas de código tienen el mayor impacto en defectos de código fuente. Plantearon las siguientes preguntas mediante el estudio de cinco aplicaciones para móviles de código abierto escrito para la plataforma Android. En el cual se encontró que: los archivos de código fuente de defectos propensos tienden a ser más dependientes



de la plataforma Android de archivos de código fuente sin defectos, el aumento de la dependencia de la plataforma aumenta la probabilidad de encontrar un defecto en un archivo de código fuente. Por el momento, los desarrolladores que buscan priorizar sus esfuerzos de aseguramiento de la calidad del software primero deben examinar los archivos de código fuente con los ratios más altos de dependencia plataforma.

En el año 2015, (Sanjuán et al., 2015), realizó un estudio de *“Comparación de dos tecnologías de desarrollo de aplicaciones móviles desde la perspectiva del rendimiento como atributo de calidad”* Universidad Tecnológica de Pereira, Colombia. La investigación tiene por objetivo comparar las dos tecnologías utilizadas en el desarrollo de aplicaciones móviles: tecnología de desarrollo nativo y tecnología basada en web, desde la perspectiva del atributo de calidad del rendimiento, haciendo uso de estándares internacionales ISO/IEC 14598 e ISO/IEC 9126. Desarrollaron una aplicación que se diseñó para ser instalada en un Smartphone Android con sistema operativo 4.0 Ice CreamSandwich. La aplicación consiste en un Really Simple Syndication (RSS), que es un sencillo lector o recopilador de contenidos que presenta cinco fuentes, de las cuales se puede tener acceso a las últimas noticias. La aplicación nativa se realizó con el kit de desarrollo de software (SDK) de Android y el IDE Eclipse con el plugin ADT. La aplicación basada en web se realizó con Phonegap porque es la plataforma más utilizada por los desarrolladores, además es compatible con jQuery Mobile, que fue el framework seleccionado para la construcción de interfaces gráficas. La metodología para el desarrollo de la aplicación se hizo uso del modelo ágil Extreme Programming (XP). Se tomaron medidas de comportamiento temporal (Tiempo de respuesta) y utilización de recursos (CPU, memoria RAM, batería) mediante el uso de la herramienta Little Eye, instalada en una computadora de escritorio. Los resultados de esta evaluación sugieren que: el tiempo de respuesta es menor en la aplicación nativa que en la versión basada en web y la utilización de recursos tanto de memoria



como de CPU y de batería, también es menor en la versión nativa de la aplicación móvil. En cuanto a la utilización de batería, cabe resaltar que el uso de este recurso en ambas versiones de la aplicación es el que presenta una diferencia más pequeña.

(Kundu & Paul, 2016) “*Improving Android performance and energy efficiency*”. En esta investigación se ha presentado los beneficios de usar el rendimiento de Android con popular Sun incorporado JVM ejecutando en la parte superior más eficiente de la energía mediante el mejoramiento del rendimiento de las aplicaciones JAVA. Desarrollaron un framework JAVA DSP que permite a las aplicaciones JAVA de Android usar ARM y DSP paralelamente y así mejorar el rendimiento. También demostraron que Android se puede hacer más eficiente desde el punto de vista energético utilizando un framework de desarrollado. Teniendo como conclusión que no todas las aplicaciones pueden beneficiarse del paralelismo. Algunas aplicaciones necesitan más de dos procesadores para mejorar el rendimiento. Además, ARM & DSP tiene diferentes velocidades de reloj y diferente tamaño de caché. Este análisis mostró que una clase de aplicaciones de Android puede beneficiarse de la utilización de nuestro marco desarrollado. Esta clase de aplicaciones de android debe ser CPU intensiva y debe ser capaz de alcanzar una velocidad significativa de dos procesadores teóricamente. Cuanto más nos aceleramos, más mejoraremos la eficiencia energética de las aplicaciones Android. Por lo tanto, el uso inteligente del framework mejorará el consumo de energía de Android.

## 2.3. Base teórica científicas

### 2.3.1. Aplicaciones móviles

(Cuello & Vittone, 2014) Las aplicaciones móviles, están presentes en los teléfonos desde hace tiempo; de hecho, ya estaban incluidas en los sistemas operativos de Nokia o Blackberry años atrás. Los móviles de esa época contaban con pantallas reducidas y muchas veces no táctiles, y son





los que ahora llamamos feature phones, en contraposición a los Smartphones, más actuales.

Actualmente encontramos aplicaciones de todo tipo, forma y color, pero en los primeros teléfonos, estaban enfocadas en mejorar la productividad personal: se trata de alarmas, calendarios, calculadoras y clientes de correo.

Hubo un cambio grande con el ingreso de iPhone al mercado, ya que con él se generaron nuevos modelos de negocio que hicieron de las aplicaciones algo rentable, tanto para desarrolladores como para los mercados de aplicaciones, como App Store, Google Play y Windows Phone Store.

Al mismo tiempo, también mejoraron las herramientas de las que disponían diseñadores y programadores para desarrollar apps, facilitando la tarea de producir una aplicación y lanzarla al mercado incluso por cuenta propia.

#### **2.3.1.1. Tipos de aplicaciones según su desarrollo**

A nivel de programación, existen varias formas de desarrollar una aplicación. Cada una de ellas tiene diferentes características y limitaciones, especialmente desde el punto de vista técnico. (Cuello & Vittone, 2014)

##### **2.3.1.1.1. Aplicaciones nativas**

(Cuello & Vittone, 2014) Son aquellas que han sido desarrolladas con el software que ofrece cada sistema operativo a los programadores, llamado genéricamente Software Development Kit o SDK. Así, Android, iOS y Windows Phone tienen uno diferente y las aplicaciones nativas se diseñan y programan específicamente para cada plataforma, en el lenguaje utilizado por el SDK.

Las aplicaciones nativas se actualizan frecuentemente y en esos casos, el usuario debe volver a descargarlas para obtener la última versión, que a veces corrige errores o añade mejoras.

Una característica generalmente menospreciada de las apps nativas es que pueden hacer uso de las notificaciones del sistema operativo para mostrar avisos importantes al usuario, aun cuando



no se esté usando la aplicación, como los mensajes de WhatsApp, por ejemplo.

Además, no requieren Internet para funcionar, por lo que ofrecen una experiencia de uso más fluida y están realmente integradas al teléfono, lo cual les permite utilizar todas las características de hardware del terminal, como la cámara y los sensores (GPS, acelerómetro, giróscopo, entre otros). A nivel de diseño, esta clase de aplicaciones tiene una interfaz basada en las guías de cada sistema operativo, logrando mayor coherencia y consistencia con el resto de aplicaciones y con el propio SO. Esto favorece la usabilidad y beneficia directamente al usuario que encuentra interfaces familiares.

#### **2.3.1.1.2. Aplicaciones Web**

(Cuello & Vittone, 2014) La base de programación de las aplicaciones web (webapps), es el HTML, conjuntamente con JavaScript y CSS, herramientas ya conocidas para los programadores web. En este caso no se emplea un SDK, por eso, estas aplicaciones pueden ser fácilmente utilizadas en diferentes plataformas sin mayores inconvenientes y sin necesidad de desarrollar un código diferente para cada caso particular.

Las aplicaciones web no necesitan instalarse, ya que se visualizan usando el navegador del teléfono como un sitio web normal. Por esta misma razón no se distribuyen en una tienda de aplicaciones, sino que se comercializan y promocionan de forma independiente. Al tratarse de aplicaciones que funcionan sobre la web, no es necesario que el usuario reciba actualizaciones, ya que siempre va a estar viendo la última versión. Pero a diferencia de las apps nativas, requieren de una conexión a internet para funcionar correctamente.

Adicionalmente tienen algunas restricciones e inconvenientes en factores importantes como gestión de memoria y no permiten



aprovechar al máximo la potencia de los diferentes componentes de hardware del teléfono.

Las aplicaciones web suelen tener una interfaz más genérica e independiente de la apariencia del sistema operativo, por lo que la experiencia de identificación del usuario con los elementos de navegación e interacción suele ser menor que en el caso de las nativas.

### **2.3.1.1.3. Aplicaciones Híbridas**

(Cuello & Vittone, 2014) Este tipo de aplicaciones es una especie de combinación entre aplicaciones nativas y web. La forma de desarrollarlas es parecida a la de una aplicación web, usando HTML, CSS, y JavaScript, y una vez que la aplicación está terminada, se compila o empaqueta de forma tal, que el resultado final es como si se tratara de una aplicación nativa. Esto permite casi con un mismo código obtener diferentes aplicaciones, por ejemplo, para Android y iOS y distribuirlas en cada una de sus tiendas. A diferencia de las aplicaciones web, estas permiten acceder, usando librerías a las capacidades del teléfono, tal como lo haría una app nativa.

Las aplicaciones híbridas, también tiene un diseño visual que no se identifica en gran medida con el del sistema operativo. Sin embargo, hay formas de usar controles y botones nativos de cada plataforma para apegarse más a la estética propia de cada una.

Existen algunas herramientas para desarrollar este tipo de aplicaciones Apache Cordova es una de las más populares, pero hay otras como Icenium, que tienen la misma finalidad. (Cuello & Vittone, 2014)



### **2.3.2. Entornos de desarrollo**

(Luna, 2014) Los entornos de desarrollo (IDE, por sus siglas en inglés) permiten incorporar funcionalidades extra de manera nativa para ejecutar los desarrollos en un ambiente de test, simulador y hasta para compilar los proyectos realizados de manera nativa.

Dentro de los entornos de desarrollo más comunes encontramos a Dreamweaver, Visual Studio, QNX Momentics IDE, Titanium Studio, entre otros. Los frameworks también permiten realizar compilaciones de proyectos que se desarrollan de manera nativa, pero generalmente requieren de un plugin que permita llevar a cabo esta acción.

### **2.3.3. Plataformas móviles**

#### **2.3.3.1. Android**

(Valenzuela, 2012) Es un sistema operativo para móviles que fue construido sobre Linux Kernel 2.6, está pensado para plataformas móviles.

Android fue lanzado bajo la licencia Apache, una licencia libre y de código abierto, lo que significa que cualquiera que quiera usar Android puede hacerlo descargando el código fuente. Por otra parte, los fabricantes de hardware pueden agregar sus propias extensiones propietarias. Este modelo de consumidores de aplicaciones móviles. Actualmente se incrementó el uso de dispositivos móviles como son celulares, tabletas, etc.

La principal ventaja de la adopción de Android es que ofrece un enfoque unificado para el desarrollo de aplicaciones. Los desarrolladores solo necesitan desarrollar para Android, y sus aplicaciones deben ser capaces de funcionar con numerosos dispositivos diferentes, siempre y cuando lo más importante de la cadena de éxito. Los fabricantes de dispositivos, por lo tanto, ven a Android como su mejor esperanza para desafiar a la embestida de los iPhone, que ya goza de una gran base de aplicaciones.



(Cuello & Vittone, 2014) Las aplicaciones de Android se programan en Java haciendo uso de librerías propias de Android, por lo que, a nivel de programación, un desarrollador con conocimientos sólidos de Java estándar no debería tener demasiados problemas para empezar a ser parte de la vida del mundo androide.

En el momento del desarrollo, Android Studio permite usar los simuladores de diferentes dispositivos, o conseguir una prueba de funcionamiento más real conectando el terminal al ordenador.

### 2.3.3.1.1. Arquitectura Android

(Benbourhala, 2015) La elección de la arquitectura Android se basa en la idea de controlar los recursos y el consumo. Las aplicaciones Android se ejecutan en un sistema con restricciones (memoria disponible, consumo de batería, diferencias en la visualización, almacenamiento disponible).

Como desarrollador, deberá prestar especial atención a los siguientes puntos:

- La creación de nuevos objetos.
- El uso de recursos (procesador, RAM, almacenamiento, etc.).
- El consumo de la batería.
- La diversidad de tamaños y resoluciones de pantalla y de configuraciones de hardware.
- La diversidad de versiones de Android disponibles en el mercado.

La arquitectura Android se compone de cinco partes diferenciadas:

- **Aplicación:** representa el conjunto de aplicaciones proporcionadas con Android.
- **Framework Android:** representa el framework que permite a los desarrolladores crear aplicaciones accediendo al conjunto de API y funcionalidades disponibles en el teléfono (fuentes de contenido, gestor de recursos, gestor de notificaciones, gestor de actividades, etc.)



- **Librerías:** Android dispone de un conjunto de librerías que utilizan los distintos componentes del sistema.
- **Android Runtime:** contiene, entre otro, la máquina virtual ART.
- **Linux Kernel:** el núcleo Linux (2.6) que proporciona una interfaz con el hardware y que gestiona la memoria, los recursos y los procesos Android.

#### 2.3.3.1.2. ART (Android RunTime)

(Benbourhala, 2015) Android se basa en una nueva máquina virtual (desde la versión 5.0 Lollipop) particular, denominada ART (Android RunTime). Reemplaza a Dalvik y posee varias características particulares:

- AOT Compilation (Ahead of Time – compilación anticipada) que permite compilar una aplicación en tiempo de instalación y no en tiempo de ejecución. Esto permite mejorar enormemente el rendimiento de las aplicaciones, puesto que no se compilan con cada ejecución.
- Mejora del Garbage Collector (recolector de basura).
- Mejora en el desarrollo y depuración de aplicaciones (los fallos ofrecen mensajes más detallados).

#### 2.3.3.1.3. NDK (Native Development Kit)

(Benbourhala, 2015) Es posible desarrollar de forma nativa en Android utilizando el NDK (Native Development Kit- kit de desarrollo nativo), que se basa en C/C++.

También puede utilizar un mecanismo Java que le permitirá llamar a código nativo desde una clase JNI (Java Native Interface).

El NDK le da acceso a:

- Un conjunto de herramientas que permiten generar código nativo desde archivos fuente C/C++.
- Una herramienta de creación de archivos apk.
- La documentación, con ejemplos y tutoriales.



#### **2.3.3.1.4. APK (Android Package)**

(Benbourhala, 2015) Un APK es el archivo binario que representa una aplicación. Este formato se utiliza para distribuir e instalar aplicaciones.

Para crear un APK, se debe compilar y empaquetar en un archivo una aplicación Android. Este archivo contendrá:

- El código de la aplicación compilada.
- Los recursos.
- Los assets.
- Los certificados.
- El archivo de manifiesto.

Para poder publicar su aplicación en Google Play, necesita crear un archivo APK.

Para generar su APK desde Android Studio, Android proporciona una herramienta que se llama aapt (Android Asset Packaging Tool – herramienta que permite crear, visualizar y modificar los archivos Android-APK), incluida en el SDK Android e integrada en el entorno de desarrollo.

Puede generar dos tipos de APK:

- Un APK firmado.
- Un APK sin firmar. Este formato puede usarse para probar su aplicación, pero no puede publicarse en Google Play.

#### **2.3.3.1.1. Ciclo de vida de una actividad**

(Benbourhala, 2015) Por defecto, cada aplicación Android se ejecuta en un proceso separado. Android gestiona los recursos disponibles en el dispositivo y puede, si fuera necesario, cerrar aplicaciones para liberar recursos (excepto aplicaciones en ejecución).

La elección de la aplicación que se cerrará depende fundamentalmente del estado del proceso en el que se encuentra. Si Android debe elegir entre dos aplicaciones que se encuentran en el



mismo estado, elegirá que se encuentre en este estado desde hace más tiempo.

El ciclo de vida de una actividad es bastante complejo y su comprensión es indispensable en el desarrollo para Android. El siguiente esquema resume este ciclo de vida:

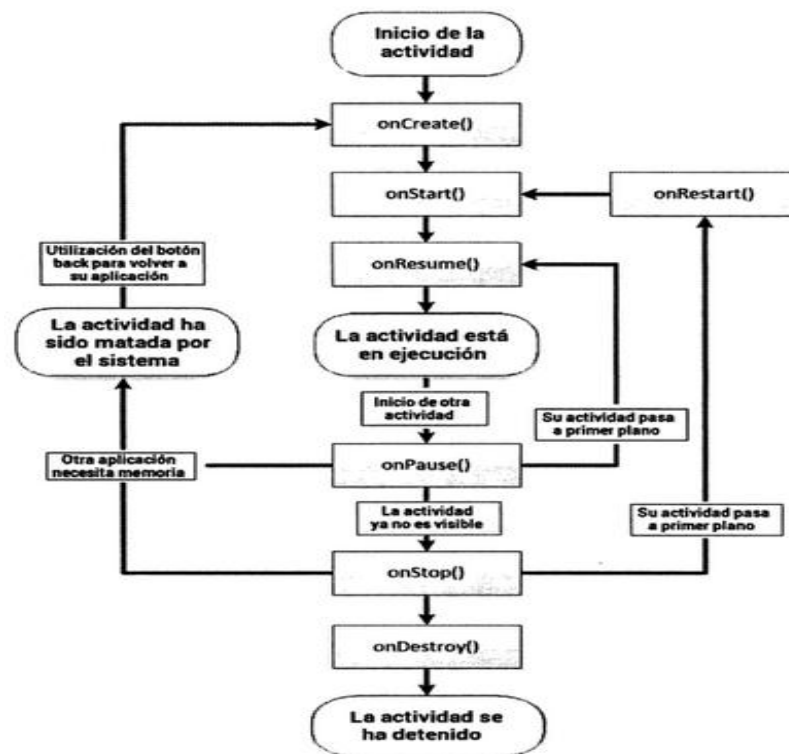


Ilustración 1 Ciclo de vida de una Actividad Android  
Fuente: (Benbourhala, 2015)

Cuando se inicia la actividad, se invoca al método onCreate. En este método, debe inicializarse su vista y vincular los datos a una lista. Este método recibe como parámetro un Bundle que contiene el estado anterior de la actividad.

A esta llamada le sigue el método onStart, que permite indicar el inicio efectivo de la aplicación.

A continuación, se invoca al método onResume para ejecutar todos los tratamientos necesarios para el funcionamiento de la actividad (thread, proceso, tratamiento), inicializar variables y listeners. Estos tratamientos deberán detenerse cuando se invoque al método





onPause y relanzarse, si fuera necesario, cuando se produzca una futura llamada al método onResume.

Después de estas tres llamadas, la actividad se considera utilizable y puede recibir interacciones de los usuarios.

Si otra actividad pasa a primer plano, la actividad en ejecución pasará a estar pausada. Justo antes de la llamada al método onPause, se invocará al método onSaveInstanceState para permitirle guardar los datos importantes de la actividad. Estos datos podrán aplicarse a futuras ejecuciones de la actividad con la llamada al método onRestoreInstanceState o la llamada a onCreate.

El método onPause permite detener los tratamientos realizados (tratamientos no Cesario si la actividad no es visible) por la actividad (tratamiento, thread, proceso).

Si su actividad pasa a estar visible de nuevo, se realizará una llamada al método onResume.

El paso de la actividad al estado “parada” tiene asociado una llamada al método onStop. En este método hay que detener todos los tratamientos restantes.

Una vez parada, su actividad puede:

- **Volver a iniciarse:** acción realizada por una llamada al método onStart siguiendo el ciclo de vida normal de la actividad.
- **Terminar:** se realiza con una llamada al método onDestroy, en el que deberá parar todos los tratamientos restantes, cerrar todas las conexiones a la base de datos, todos los threads, todos los archivos abiertos, etc. Puede provocar la llamada al método onDestroy utilizando el método finish.

### 2.3.3.2. IOS

(Cuello & Vittone, 2014) Un programador que quiera empezar a hacer magia desarrollando para iPhone e iPad debe tener una base de programación orientada a objetos, algo que le permitirá luego una transición más transparente a Objective-C, el lenguaje de programación que se usa en estos casos.



A nivel de hardware y software para desarrollar aplicaciones para iOS, se necesita un ordenador Mac con el Kit de Desarrollo de Software (SDK), que en este caso es Xcode, el software oficial de Apple para desarrollo para iPhone e iPad, de descarga gratuita.

El código se puede probar directamente en el simulador, una representación del teléfono que permite ver cómo se comporta el código dentro del ordenador, algo que sirve para la mayoría de los casos, pero tiene ciertas limitaciones y no es completamente fidedigno, pues suele comportarse más rápido de que realmente lo hace en el teléfono.

Idealmente, para hacer una prueba más real de desarrollo, hay que probar el código en un iPhone conectado al Mac. Para esto es necesario pagar una licencia de desarrollador, algo que, de todas formas, hará falta más adelante para publicar la aplicación en la tienda.

### **2.3.3.3. Windows Phone**

(Cuello & Vittone, 2014) Un programador que haya estado trabajando en C# va tener un buen comienzo, ya que este es el lenguaje que, junto con las librerías propias de Windows Phone, hacen una aplicación posible.

Quien no sea muy fanático de Windows no va a tener otra opción que usar, por lo menos, Windows 7 en su ordenador para desarrollar. El consuelo es que puede instalarse tanto en un PC como en la máquina virtual de un Mac. Y hablando de software, también hace falta tener Microsoft Visual Studio. La versión gratuita es suficiente para desarrollar apps, pero está claro que pagando una versión completa se tiene acceso a muchas más comodidades.

El simulador de Windows Phone es relativamente bueno, porque la app se puede probar directamente desde el ordenador. Para tener una simulación más real, es posible conectar el teléfono al ordenador



con Windows, si se tiene una licencia de desarrollo que cuesta 99 dólares anuales.

#### **2.3.4. Librerías y Frameworks Móviles**

(Luna, 2014) Una librería es un conjunto de tecnologías que puede englobar características de CSS y JavaScript y que nos facilita, de alguna manera, el desarrollo de una solución web para ambas plataformas. Dentro de las librerías más conocidas, podemos mencionar a jQuery Mobile, Sencha Touch y jQuery UI, entre otras.

Estas se ocupan de compactar funcionalidades que requieren mucho tiempo de elaboración por parte de un desarrollador en funciones específicas que nos aportan agilidad al momento de desarrollar y nos permiten despreocuparnos por la estética o solución cuando el proyecto deba ejecutar en diferentes plataformas.

El framework nos permite englobar, en un único entorno, todo el conjunto de archivos y APIs que nos permiten desarrollar una solución, estructurados de una manera jerárquica.

Dentro de los frameworks más conocidos, podemos mencionar a Eclipse y Netbeans como los más populares y gratuitos, y a Dreamweaver dentro de los frameworks de pagos. Todos estos prestan características similares, aunque con algunos toques personales, y permiten incorporar las librerías mencionadas anteriormente sin mayores problemas.

#### **2.3.5. Calidad de Software**

(Coral Calero Muñoz, 2010 ) El modelo ISO/IEC 9126, para la calidad del producto software, publicado en 1991 y revisado en 2001, está siendo incorporado en la nueva serie ISO/IEC 25000 (SQuaRe: Software Quality Requirements, Requisitos y Evaluación de la calidad de productos software) ISO/IEC 25000 (2009).

Esta serie de estándares interpretan la calidad de un sistema software como el grado en el que el sistema satisface implícitas y explícitas de sus diferentes usuarios. Estas necesidades se representan dentro de SQuaRe



en diferentes modelos: el modelo de calidad del producto software, el modelo de calidad de datos y el modelo de calidad en uso del sistema.

La principal finalidad del modelo de calidad del producto software es especificar y evaluar la calidad de los productos software, ya sea a través de medidas de internas, directas de las propiedades inherentes del software o mediante medidas externas, indirectas del comportamiento del sistema del que forma parte. Las medidas externas son también utilizables para especificar y evaluar la calidad de ciertos aspectos del sistema de computación completo (hardware y software).

### 2.3.6. Factores de Calidad de Software

(Juan Pablo Carvallo) El estándar ISO/IEC 9126 características de calidad de un producto software, define las características y sub-características de calidad de los productos software. Está basada en la terminología de calidad recogida en el estándar ISO 8402.

También define seis características principales que cubren todos los aspectos de la calidad del software. Cada una de estas características se descompone en una serie de sub-características. A continuación, se recogen las características principales:

1. **Funcionalidad:** capacidad de un producto software para funciones que cubran las necesidades explícitas e implícitas cuando el software es utilizado bajo las condiciones específicas.
2. **Fiabilidad:** capacidad del producto software para mantener un nivel especificado de rendimiento cuando es usado bajo las condiciones especificadas.
3. **Usabilidad:** Capacidad de un producto software para ser entendido, aprendido, usado y atractivo para el usuario, cuando es usado bajo las condiciones especificadas.
4. **Eficiencia:** capacidad de un producto software para proporcionar un rendimiento adecuado, relativo a la cantidad de recursos utilizados, bajo unas condiciones establecidas.



5. **Mantenibilidad:** Capacidad de un producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptaciones del software a cambios en el entorno, en los requerimientos o las especificaciones funcionales.
6. **Portabilidad:** Capacidad del producto software para ser transferido de un entorno a otro.

### 2.3.7. Métricas de Calidad

Se conoce con el nombre de métrica a las medidas numéricas que se efectúan sobre productos y procesos del desarrollo de software. Para cada factor de calidad debe fijarse una métrica. Los estándares ISO/IEC 14598 e ISO/IEC 9126 proporcionan para evaluar la calidad de un producto software.

### 2.3.8. Norma ISO/IEC 25000

ISO/IEC 25000, conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software. La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones. (ISO 25000, 2015)

### 2.3.9. ISO/IEC 25010

El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software



determinado. La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas. (ISO 25000, 2015)

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura.



Ilustración 2 Características ISO 25000  
Fuente: ("ISO 25010," 2017)

### Eficiencia de desempeño

Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

**Comportamiento temporal.** Los tiempos de respuesta y procesamiento y los ratios de *throughput* de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas (*benchmark*) establecido.



**Utilización de recursos.** Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.

**Capacidad.** Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

## 2.4. Definición de términos básicos

- a) **Android:** es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tablets o tabléfonos; y también para relojes inteligentes, televisores y automóviles.
- b) **Rendimiento:** Capacidad del producto software para proporcionar prestaciones apropiadas, relativas a la cantidad de recursos usados, bajo condiciones determinadas.
- c) **Frameworks:** El framework nos permite englobar, en un único entorno, todo el conjunto de archivos y APIs que nos permiten desarrollar una solución, estructurados de una manera jerárquica.
- d) **Aplicación móvil:** es un programa que usted puede descargar y al que puede acceder directamente desde su teléfono o desde algún otro dispositivo móvil – como por ejemplo una tablet o un reproductor MP3.



## CAPÍTULO III: MARCO METODOLÓGICO

### 3.1. Tipo y diseño de investigación

#### 3.1.1. Tipo de investigación

El presente trabajo corresponde a una investigación de tipo Cuantitativa – Aplicada porque consiste en utilizar la recolección y el análisis de datos para contestar preguntas de investigación y probar hipótesis establecidas previamente, y confía en la medición numérica, el conteo y frecuentemente en el uso de la estadística.

#### 3.1.2. Diseño de la investigación

De acuerdo con el tipo de investigación el diseño utilizado es cuasi experimental porque el investigador manipula la variable independiente para ver su efecto y relación con una a más variables dependientes. Los sujetos no son asignados al azar, ni emparejados, sino que dichos grupos ya estaban formados antes del experimento.

### 3.2. Población y muestra

#### 3.2.1. Población

La población está definida por top 9 de frameworks de desarrollo móvil más usados, el cual ha sido seleccionado por conveniencia.

#### 3.2.2. Muestra

La muestra ha sido definida por conveniencia siendo dos frameworks de desarrollo móvil, los cuales fueron seleccionados de acuerdo a una evaluación.

### 3.3. Hipótesis

El framework Apache Cordova es el más óptimo en cuanto a rendimiento.

### 3.4. Variables

#### 3.4.1. Variable dependiente

Rendimiento de aplicaciones móviles Android.

#### 3.4.2. Variable independiente

Frameworks de desarrollo.





### 3.5. Operacionalización

VARIABLE	DEFINICION CONCEPTUAL	CARACTERISTICA	SUBCARACTERISTICAS	METRICAS	FORMULA POR PRUEBA	FÓMULA GENERAL	CRITERIO
<p><u>DEPENDIENTE</u></p> <p><i>Rendimiento en aplicaciones móviles Android.</i></p>	<p>Una aplicación Android se compone de una estructura general; conformada por librerías de código, archivos de recursos y vistas, código fuente y Android Manifest.</p>	<p>Eficiencia en el desempeño</p>	<p>Comportamiento en el tiempo</p>	<p>tiempo de respuesta(s)</p>	<p>R=B - A</p> <p>A=Tiempo de envío de petición</p> <p>B= Tiempo en recibir la primera respuesta</p>	<p><math>X = \sum R/n</math></p> <p>N=número de pruebas</p>	<p>0 &lt; X El más cercano a 0 es el mejor.</p>
				<p>tiempo de espera(s)</p>	<p>R = B - A</p> <p>A=Tiempo cuando se inicia un trabajo</p> <p>B=Tiempo en completar el trabajo</p>	<p><math>X = \sum R/n</math></p> <p>N=número de pruebas</p>	
			<p>Rendimiento</p>	<p>R=A/T</p> <p>A=Número de tareas completadas</p> <p>T=Intervalo de tiempo Donde: T&gt;0</p>	<p><math>X = \sum R/n</math></p> <p>N=número de pruebas</p>	<p>0 &lt; X El más lejano a 0/t es el mejor</p>	
			<p>Uso de CPU (%)</p>	<p>R = A</p> <p>A= cantidad de CPU que es usado para realizar una tarea</p>	<p><math>X = \sum R/n</math></p> <p>N=número de pruebas</p>		
			<p>Utilización de recursos</p>	<p>Uso de memoria RAM (MB)</p>	<p>R = A</p> <p>A= Cantidad de memoria que es usado para realizar una tarea</p>	<p><math>X = \sum R/n</math></p> <p>N=número de pruebas</p>	<p>0&lt;=X El más cercano a 0 es el mejor.</p>



## **3.6. Abordaje metodológico, técnicas e instrumentos de recolección de datos**

### **3.6.1. Abordaje metodológico**

El presente trabajo de investigación utilizó como método de recolección de datos la Observación, la cual sirvió como instrumento de análisis, La observación científica "tiene la capacidad de describir y explicar el comportamiento, al haber obtenido datos adecuados y fiables correspondientes a conductas, eventos y /o situaciones perfectamente identificadas e insertas en un contexto teórico.

### **3.6.2. Técnicas de recolección de datos**

#### **Observación**

Es una técnica que consiste en observar atentamente el comportamiento de la aplicación, tomar información y registrarla para su posterior análisis. La observación es un elemento fundamental de todo proceso investigativo; en ella se apoya el investigador para obtener el mayor número de datos. Teniendo un objetivo claro, definido y preciso.

### **3.6.3. Instrumentos de recolección de datos**

#### **Ficha de observación**

La ficha de observación es considerada como una especie de procedimiento de investigación, el cual consiste básicamente en poder utilizar instrumentos adecuados para poder establecer una relación entre la hipótesis y los hechos reales, a través de la observación científica, también de la investigación sistematizada y ordenada.

#### **Android Profiler**

Esta herramienta de perfilado proporciona datos en tiempo real para la CPU, la memoria y la actividad de la red de su aplicación. Puede llevar a cabo el método basado en muestras trazando el tiempo de



su ejecución de código, las asignaciones de vista de la memoria, e inspeccionar los detalles de los archivos en la red transmitida.

Este software se utilizó para obtener los datos referentes a CPU, memoria y actividad de la red, obteniendo además el tiempo que genera cada evento la aplicación.

### **Apowersoft**

Es una herramienta de escritorio que permite la grabación de pantalla y de audio. Permite guardar videos como archivos estándar MP4. El cual nos permite grabar los eventos que se realizan en la aplicación móvil.

## **3.7. Procedimiento para la recolección de datos**

- a El objeto a evaluar será una aplicación real desarrollada con frameworks de desarrollo seleccionados.
- b Elaborar una ficha con los criterios a observar en la aplicación.
- c Se empleó un software de monitoreo para obtener los datos de los indicadores de la aplicación móvil.
- d Elaborar un formato para registrar los datos observados por el software de monitoreo.
- e Analizar e interpretar los datos obtenidos.
- f Elaborar el informe de observación con los datos obtenidos.

## **3.8. Análisis estadístico e interpretación de los datos**

Se aplicarán los siguientes modelos estadísticos para la obtención de los indicadores de rendimiento.

### **Comportamiento en el tiempo:**

- Tiempo de respuesta
$$X = B - A$$
  - o A = Tiempo de envío de petición
  - o B= Tiempo en recibir la primera respuesta
- Tiempo de espera
$$X = B - A$$
  - o A = Tiempo cuando se inicia un trabajo



- B = Tiempo en completar un trabajo
- Rendimiento
  - $X = A/T$
  - A=Número de tareas completadas
  - T=Intervalo de tiempo Donde:  $T > 0$

#### Utilización de recursos

- Uso de CPU
  - $X = A$
  - A= cantidad de CPU que es usado para realizar una tarea
- Uso de memoria RAM
  - $X = A$
  - A= Cantidad de memoria que es usado para realizar una tarea.

### 3.9. Principios éticos

Los criterios éticos que se respetan en el presente proyecto de tesis es el Código Deontológico del Colegio de Ingenieros de Perú en su Capítulo II “De la Relación con el Público” en su artículo 106 expresa:

Los ingenieros, al explicar su trabajo, méritos o emitir opiniones sobre temas de ingeniería, actuarán con seriedad y convicción, cuidando de no crear conflictos de intereses, esforzándose por ampliar el conocimiento del público a cerca de la ingeniería y de los servicios que presta a la sociedad.

### 3.10. Criterios de rigor científico

La presente propuesta de investigación se realiza siguiendo los juicios científicos establecidos, estos permiten garantizar la calidad de la propuesta de investigación.



## CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS

### 4.1. Resultados en tablas y gráficos

En la tabla 9 se muestra específicamente los valores obtenidos de las características que se seleccionaron para ser aplicadas en la evaluación de calidad externa, donde se puede observar que la aplicación móvil desarrollada con JQuery Mobile ha obtenido mayor valor frente a Apache Cordova.

Aplicaciones Móviles	CARACTERÍSTICA	VALOR PARCIAL TOTAL (/10)	NIVEL DE IMPORTANCIA	PORCENTAJE DE IMPORTANCIA	VALOR FINAL
Apache Cordova	Eficiencia en el desempeño	287.01	A	100%	2.87
JQuery Mobile	Eficiencia en el desempeño	336.27	A	100%	3.36

Tabla 9: valor obtenido por cada aplicación móvil

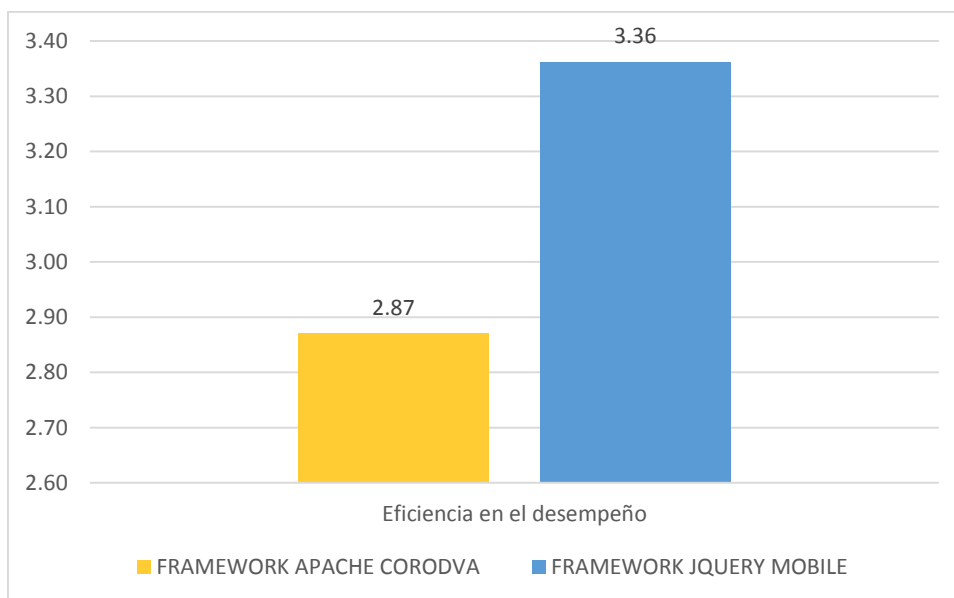
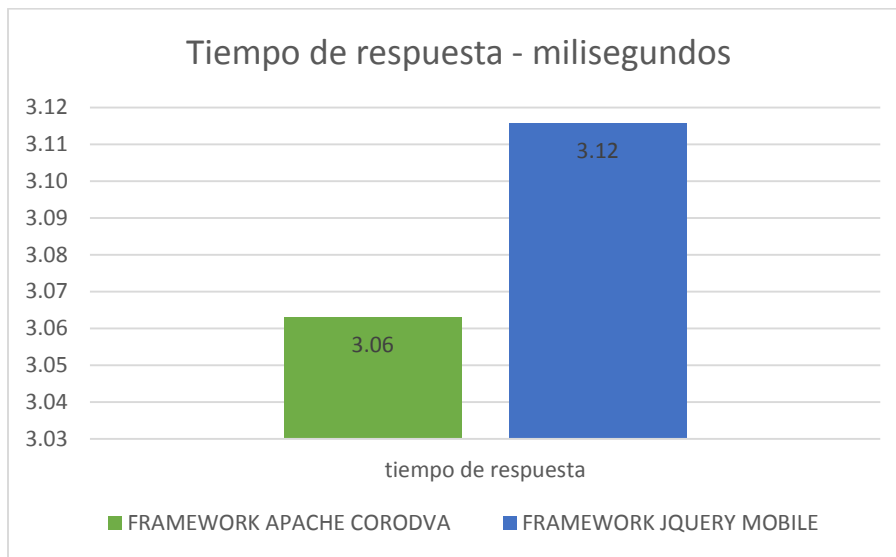


Ilustración 3 resultado eficiencia en el desempeño

Fuente: elaboración propia



### Tiempo de respuesta

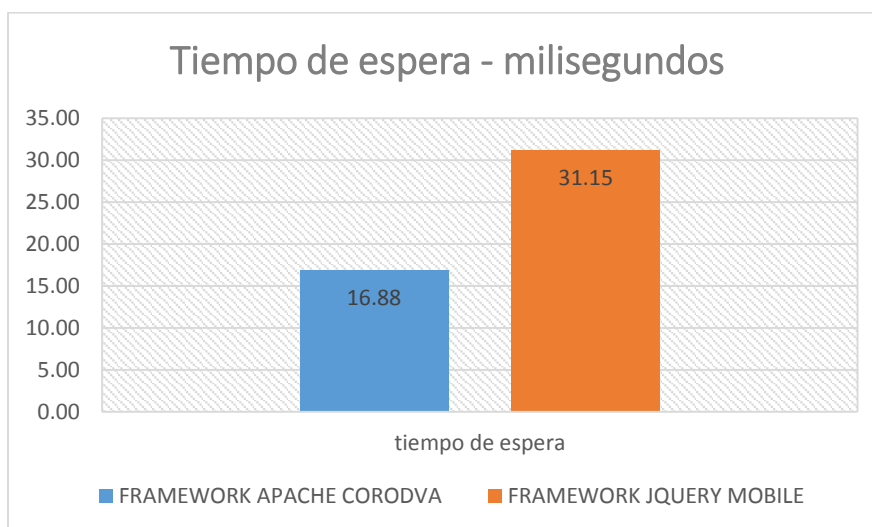


**Ilustración 4** Resultados de tiempo de respuesta

**Fuente:** Elaboración propia

La ilustración 4 muestra el tiempo promedio de respuesta de los dos frameworks, en el cual se observa que el framework Apache Cordova obtiene un valor de 3.06 y el framework jQuery Mobile con 3.12, siendo el caso que Apache Cordova es mejor porque el valor es más cercano a 0. En cuanto al framework jQuery Mobile necesitó 0.06 segundos más para obtener una respuesta.

### Tiempo de espera



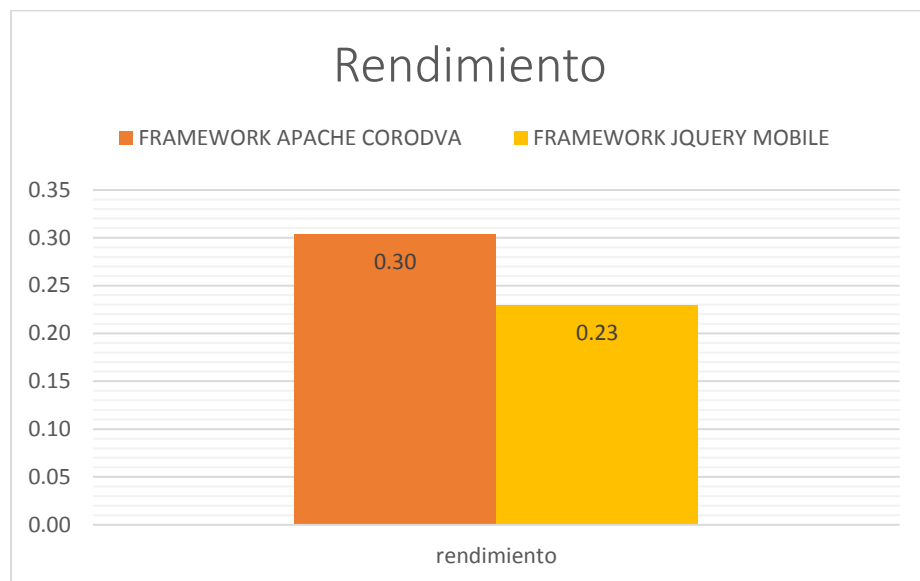
**Ilustración 5** Resultados tiempo de espera

**Fuente:** elaboración propia



La ilustración 5 muestra el tiempo promedio de espera de los dos frameworks, obteniendo 16.88 segundos el framework Apache Cordova y 31.15 segundos el framework jQuery Mobile. Considerándose que el menor valor es el más óptimo. Con una diferencia de 14.27 segundos adicionales que jQuery Mobile necesita para realizar una tarea.

**Rendimiento**

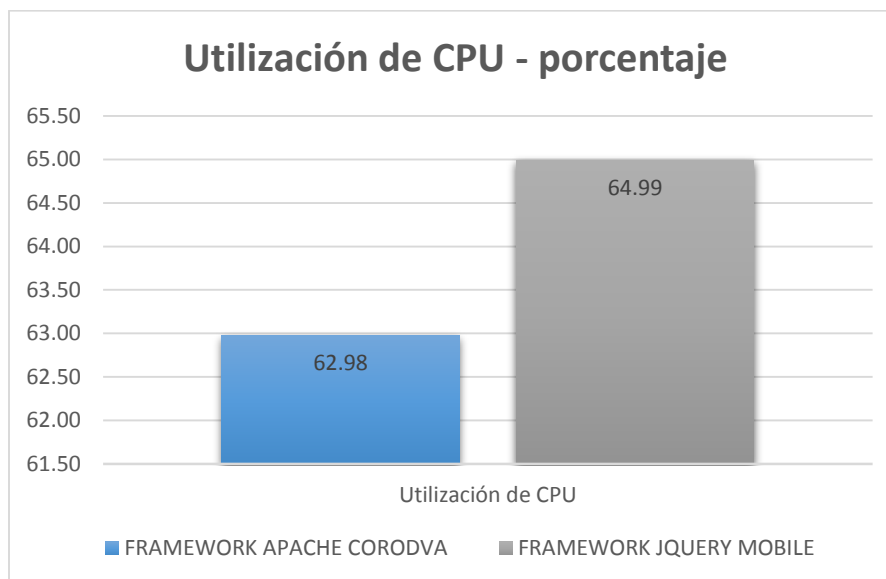


**Ilustración 6** Resultado de rendimiento  
**Fuente:** elaboración propia

La ilustración 6 muestra el rendimiento promedio de los frameworks, en el cual se observa que el framework Apache Cordova obtiene el mayor valor con 0.30 y el framework jQuery Mobile 0.23, resultando que el framework Apache cordova es el que ofrece un mejor rendimiento.



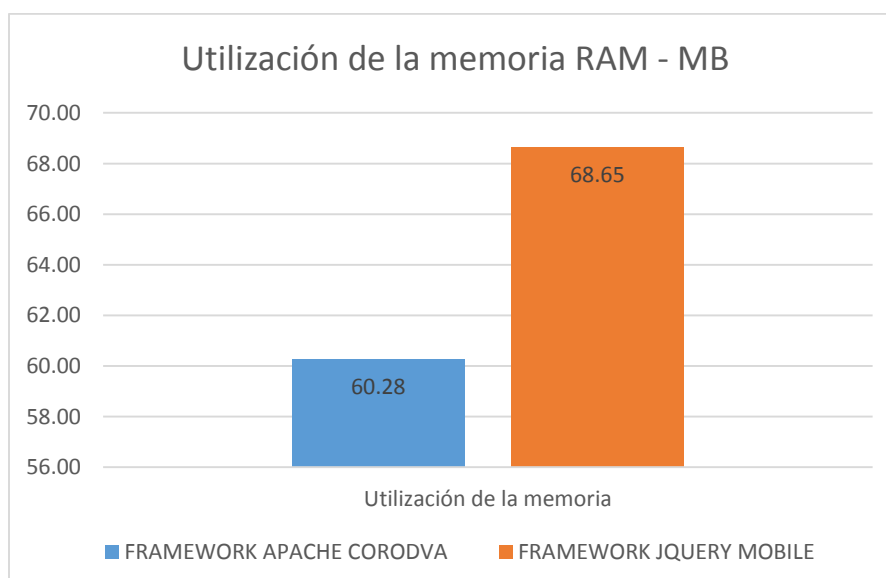
### Utilización de CPU



**Ilustración 7** Resultados de utilización de CPU  
**Fuente:** elaboración propia

De la ilustración 7, se observa que el framework Apache cordova tiene un porcentaje de 62.98 % y jQuery Mobile un 64.99%, en el cual el framework Apache córdoba es el mejor frente a jQuery Mobile con un porcentaje de diferencia de 2.01 %

### Utilización de memoria RAM



**Ilustración 8** Resultado de utilización de memoria RAM  
**Fuente:** elaboración propia





De acuerdo a la ilustración 8, se puede observar que el promedio de utilización de memoria para el framework Apache cordova es de 60.28 Mb y para el framework jQuery Mobile 68.65 Mb, siendo que el menor valor es el más óptimo, resultado ser en este caso framework Apache cordova.

#### **4.2. Discusión de resultados**

De acuerdo con los gráficos se puede observar que el framework Apache cordova es el más recomendado desde la perspectiva de eficiencia de desempeño de 2.87, siendo el más óptimo para el desarrollo de aplicaciones móviles multiplataforma. El framework Apache Cordova obtiene un 3.06ms en tiempo de respuesta, 16.88 ms en tiempo de espera, 0.30ms en rendimiento, 62.98% en utilización de CPU y 60.28MB en utilización de memoria RAM. El framework JQuery Mobile obtiene un 3.12ms en tiempo de respuesta, 31.15ms en tiempo de espera, 0.23ms en rendimiento, 64.99% en utilización de CPU y 68.65MB en utilización de memoria RAM. Considerándose de acuerdo a lo establecido por la norma ISO 2500, que el valor más cercano a 0 es el mejor.



## CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN

La propuesta que se plantea en esta investigación inicia con una evaluación preliminar para seleccionar los frameworks de desarrollo, de acuerdo a un top 9, posteriormente se definió el proceso de evaluación de acuerdo a la norma ISO 25000, que evaluará el rendimiento en las aplicaciones móviles, definiendo métricas e indicadores, para lo cual se utilizaron herramientas como ficha de observación, bitácoras y software de monitoreo de aplicaciones móviles.

Siendo los frameworks seleccionados, se procedió a su respectivo desarrollo, haciendo uso de una metodología ágil Scrum, identificando los requerimientos del cliente y arquitectura.

Una vez establecida la norma, métricas e indicadores, desarrollado las aplicaciones móviles, se realizó la medición de rendimiento de dichas aplicaciones móviles, para finalmente determinar el nivel de rendimiento.



**Ilustración 9** Diagrama de flujo que muestra el proceso de la propuesta planteada  
**Fuente:** Elaboración propia



### 5.1 Seleccionar los frameworks para el desarrollo de aplicaciones móviles.

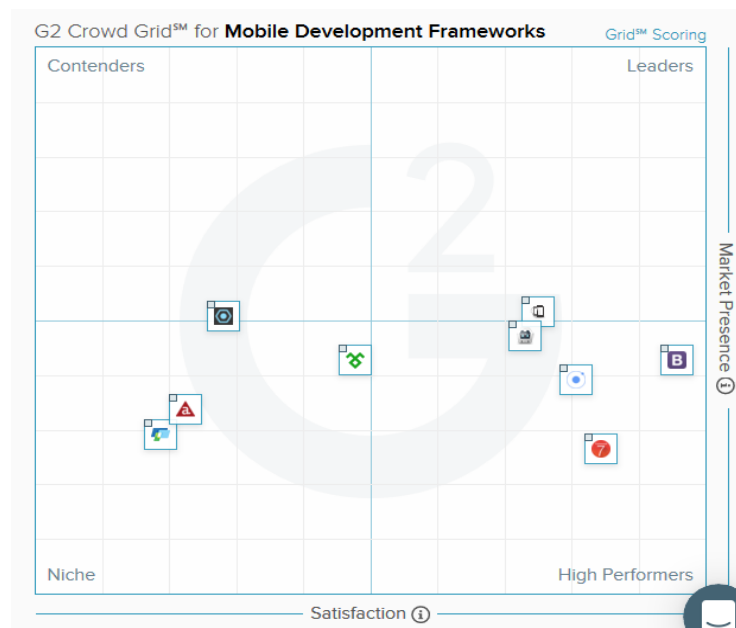
La selección de frameworks, de desarrollo de aplicaciones móviles se ha realizado mediante un modelo de evaluación, adaptada de una investigación realizada por (Guncay Barzallo & Samaniego Mosquera, 2015) Evaluación de frameworks para el desarrollo de aplicaciones móviles multiplataforma, 2015, Ecuador.

Dicha evaluación es la siguiente:

#### 1. Evaluación rápida

- a. Identificar una lista de componentes para ser evaluados:

Según estudio (G2 Crowd, 2017) El top de mejor software de frameworks de desarrollo móvil para 2017 es el siguiente:



**Ilustración 10 Cuadrante Mágico**  
**Fuente:** (G2 Crowd, 2017)

Productos en el cuadrante líder son altamente valorados por los usuarios G2 y tienen presencia en el mercado. Los líderes incluyen: PhoneGap. De alto rendimiento son muy valorados por sus usuarios, pero aún no han alcanzado la cuota de mercado y la escala de los líderes. De alto rendimiento incluyen: Apache Cordova , Ionic , Bootstrap y Framework 7. Contendientes tienen significativa presencia y recursos del mercado, pero han recibido



debajo de las calificaciones promedio de satisfacción de los usuarios o que aún no han recibido un número suficiente de comentarios para validar la solución. Los competidores incluyen: Reaccionar nativo. Soluciones de nicho no tienen la presencia en el mercado de los líderes. Es posible que hayan sido valorados positivamente en la satisfacción del cliente, pero aún no han recibido comentarios suficientes para validarlos. Productos de nicho incluyen: jQuery Mobile , Kendo interfaz de usuario , y Appcelerator Titanium. (G2 Crowd, 2017)

b. Definir filtros que permitan realizar la evaluación rápida medir cada componente

Características del filtrado inicial	Phonegap	Apache cordova	Ionic	Bootstrap	Framework 7	React Native	jQuery Mobile	Kendo UI	Appcelerator Titanium
Desarrollo multiplataforma	2	2	2	1	0	0	2	2	2
Edad del software	2	2	1	1	1	2	2	2	1
Costo del software	2	2	2	2	2	2	2	1	1
Open Source	2	2	2	2	2	1	2	1	1
Documentación	1	2	1	2	2	2	2	2	2
Popularidad	2	2	2	2	2	2	2	0	1
<b>total</b>	11	12	10	10	9	9	12	8	8

**Tabla 1** evaluación de frameworks  
 Fuente: (Guncay Barzallo & Samaniego Mosquera, 2015)

- **Desarrollo multiplataforma:** se ha considerado como multiplataforma al framework con la capacidad de desarrollar aplicaciones para más de un sistema operativo móvil. Los niveles de evaluación son los siguientes: se considera “0” cuando el número de plataformas se encuentra en el rango de 1 a 3. “1” cuando el rango de plataformas esta entre 4 a 6, y “2” cuando es mayor que 6 plataformas. (Guncay Barzallo & Samaniego Mosquera, 2015)



Multiplataforma	
Framework	Puntuación
Phonegap	2
Apache cordova	2
Ionic	2
Bootstrap	1
Framework 7	0
React Native	0
jQuery Mobile	2
Kendo UI	2
Titanium	2

**Tabla 2** evaluación multiplataforma  
**Fuente:** Elaboración propia

- Edad del software:** se ha considerado como edad idónea al framework que tenga un tiempo mayor a tres años en el mercado. Esta característica es importante para conocer la fiabilidad y también la estabilidad que puede ofrecer. Los niveles de evaluación son los siguientes: se considera “0” cuando el número de plataformas se encuentra en el rango de 1 a 2. “1” cuando el rango de plataformas esta entre 3 a 5, y “2” cuando es mayor que 6 años. (Guncay Barzallo & Samaniego Mosquera, 2015)

Edad del Software	
Framework	Puntuación
Phonegap	2
Apache cordova	2
Ionic	1
Bootstrap	1
Framework 7	1
React Native	2
jQuery Mobile	2
Kendo UI	2
Titanium	1

**Tabla 3** evaluación de edad del software  
**Fuente:** Elaboración propia



- Costo del software:** se ha considerado como viable el estudio de frameworks que se distribuyen de forma gratuita. Los niveles de evaluación son los siguientes: se considera “0” cuando framework es pagado, “1” cuando el uso de framework es limitado y “2” cuando es totalmente gratuito. (Guncay Barzallo & Samaniego Mosquera, 2015)

Costo del software	
Framework	Puntuación
Phonegap	2
Apache cordova	2
Ionic	2
Bootstrap	2
Framework 7	2
React Native	2
jQuery Mobile	2
Kendo UI	1
Titanium	1

**Tabla 4** evaluación de costo del software  
**Fuente:** Elaboración propia

- Open Source:** se consideró importante esta característica en los frameworks a evaluar, ya que un software de código abierto presenta una gran comunidad que lo respalda y contribuye a su mejora continua. Los niveles de evaluación son los siguientes: se considera “0” cuando framework parcialmente open source, “1” cuando el uso de framework es intermedio y “2” cuando es totalmente open source. (Guncay Barzallo & Samaniego Mosquera, 2015)



Open Source	
Framework	Puntuación
Phonegap	2
Apache cordova	2
Ionic	2
Bootstrap	2
Framework 7	2
React Native	1
jQuery Mobile	2
Kendo UI	1
Appcelerator Titanium	1

**Tabla 5** evaluación Open Source  
**Fuente:** Elaboración propia

- Documentación:** se ha considerado importante que los frameworks cuenten con una página oficial que ofrezca información de la herramienta; así como también manuales y tutoriales donde el desarrollador pueda orientarse y hace un correcto uso de la misma. Los niveles de evaluación son los siguientes: se considera “0” cuando la documentación de framework es básica, “1” cuando la documentación de framework es intermedio y “2” cuando la documentación del framework es avanzada. (Guncay Barzallo & Samaniego Mosquera, 2015)

Documentación	
Framework	Puntuación
Phonegap	1
Apache cordova	2
Ionic	1
Bootstrap	2
Framework 7	2
React Native	2
jQuery Mobile	2
Kendo UI	2
Appcelerator Titanium	2

**Tabla 6** evaluación de documentación  
**Fuente:** Elaboración propia



- Popularidad:** se ha considerado como popular al framework con más de un millón de coincidencias en Google (ver anexo 1), esta característica es importante puesto que un framework con una fuerte adopción presenta una gran cantidad de información de la comunidad de desarrolladores que ayudan a resolver problemas durante el uso de la herramienta. Los niveles de evaluación son los siguientes: se considera “0” cuando obtiene 1 millón de coincidencias, “1” cuando obtiene 3 millones de coincidencias y “2” obtiene más de 5 millones de coincidencias. (Guncay Barzallo & Samaniego Mosquera, 2015)

Popularidad	
Framework	Puntuación
Phonegap	2
Apache cordova	2
Ionic	2
Bootstrap	2
Framework 7	2
React Native	2
jQuery Mobile	2
Kendo UI	0
Appcelerator Titanium	1

**Tabla 7** evaluación de popularidad  
**Fuente:** Elaboración propia

De acuerdo a la evaluación de frameworks, se ha seleccionado Apache Cordova y JQuery Mobile ya que obtienen el mayor puntaje, para el desarrollo de las aplicaciones móviles.

La información de cada framework se encuentra en anexo 1.

## 5.2 Seleccionar los parámetros con el cual se evaluará la aplicación móvil.

Calidad Externa: Esta es medida y evaluada a través de propiedades dinámicas del código ejecutable de un software, es decir, cuando un módulo o la aplicación terminada es puesta en marcha en una computadora o en una red simulando a un ambiente real. En fases finales del ciclo de vida del software (en distintas etapas de testeo o ya en estado





operativo de un producto o aplicación Web), es posible medir, evaluar y controlar la calidad externa de estos productos ejecutables.(BOLIVAR MUÑOZ, 2016)

### 5.2.1 Definir el proceso de evaluación

El proceso de evaluación se definió usando la norma ISO/IEC 25000 en lo que respecta a calidad externa, con la que se determinó las características y subcaracterísticas apropiadas para la evaluación logrando obtener resultados concretos, para ello se tomó en cuenta el grado de importancia, la ponderación en porcentaje de las subcaracterísticas de calidad según la norma ISO/IEC 25000 en lo que respecta a calidad externa para estimar el porcentaje de rendimiento de las aplicaciones desarrolladas. Referente al modelo de evaluación de calidad de producto software, se procede a realizar lo siguiente:

#### Definición de las tareas

Para la definición de las tareas se ha tomado en cuenta la investigación de (Sanjuán et al., 2015) en la cual definen realizar 5 tareas para cada prueba. En base a dicha investigación se consideró lo siguiente:

TAREAS	DESCRIPCIÓN
Logueo	Para el ingreso a la aplicación móvil, deberá ingresar un nombre de usuario y contraseña registrada en la base de datos.
Crear registro	Para realizar esta operación, ingresar los datos en los campos correspondientes, seleccionar guardar.
Listar registro	Los registros se listan en una tabla, al seleccionar el botón categoría.
Editar registro	Seleccionar un registro, modificar los datos requeridos, y guardar.
Eliminar registro	Seleccionar un registro, aceptar la confirmación.

**Tabla 8** descripción de tareas  
Fuente: elaboración propia



### 5.2.2 Selección de subcaracterísticas y atributos de calidad para APPEAT

Nivel de importancia	Simbología	% Referencial del nivel de importancia	Significado
Alto	A	70% - 100%	El grado de importancia de la característica y subcaracterísticas es alto por ende se realizará las mediciones
Medio	M	25% - 69%	La característica y subcaracterísticas no es tan relevante, pero puede o no ser medida dependiendo el criterio del evaluador
Bajo	B	1% - 24%	La característica y subcaracterísticas no tiene relevancia y no será medida
No Aplica	NA	0%	Este valor se dará a la característica y subcaracterísticas que no se pueden medir dependiendo de diferentes factores

**Tabla 9** Nivel de importancia  
Fuente: (Amparo, 2014)

Para definir las subcaracterísticas y atributos de calidad interna, externa más importantes para el producto software, se determina el nivel de importancia.

**Tabla 10** subcaracterísticas y atributos de calidad externa seleccionadas

SUBCARACTERÍSTICA Y ATRIBUTOS DE CALIDAD EXTERNA				
Características	Subcaracterísticas	Nivel de importancia	Motivo de selección	Ponderación
Eficiencia en el desempeño	Comportamiento temporal	A	Se califica con valor de importancia A porque es muy necesario evaluar si la aplicación móvil proporciona los tiempos de respuesta apropiados.	100%
	Utilización de recursos	A	Se califica con valor de importancia B porque es muy necesario evaluar si la aplicación móvil utiliza los recursos adecuados mientras está operando.	
	Capacidad	B	Se califica con valor de importancia B porque no es necesario evaluar.	

**Tabla 10** subcaracterísticas y atributos de calidad externa seleccionadas  
Fuente: (Amparo, 2014)



**Métricas de calidad externa seleccionadas para la aplicación móvil ADP EAT**

En relación a la tabla 10, las métricas seleccionadas para evaluar la calidad externa de la aplicación móvil ADPEAT se especifican en la tabla 3:

MÉTRICAS PARA LA CALIDAD EXTERNA			
Característica	Subcaracterísticas	Métricas	Significado
Eficiencia en el desempeño	Comportamiento temporal	Tiempo de respuesta	El tiempo estima para completar una tarea
		Tiempo de espera	El tiempo en completar un trabajo completo con el sistema
		Rendimiento	La cantidad de tareas que pueden ser procesadas
	utilización de recursos	Utilización de CPU	La cantidad de tareas que puede ser procesadas
		Utilización de la memoria	El espacio de memoria que se utiliza para realizar una tarea

**Tabla 11 métricas para calidad externa**  
**Fuente:** (Amparo, 2014)



Sub características	Métrica	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Recursos utilizados
Comportamiento del tiempo	Tiempo de respuesta	¿Cuál es el tiempo estimado para completar una tarea?	Tomar el tiempo desde que se envía la petición hasta obtener la respuesta	$X = ? R/n$ N=número de pruebas	$\leq X \leq 1$ El más cercano a 0 es el mejor. Donde el peor caso es $\geq 15t$ .	Tester
	Tiempo de espera	¿Cuál es el tiempo desde que se envía una instrucción, para que inicie un trabajo, hasta que lo completa?	Tomar el tiempo cuando se inicia un trabajo y el tiempo en completar el trabajo	$X = ? R/n$ N=número de pruebas	$\leq X \leq 1$ El más cercano a 0 es el mejor. Donde el peor caso es $\geq 15t$ .	Tester
	Rendimiento	¿Cuántas tareas pueden ser procesadas por unidad de tiempo?	Contar el número de tareas completadas en un intervalo de tiempo	$X = ? R/n$ N=número de pruebas	$X = A/T$ El más lejano a 0/t es el mejor. Donde el mejor caso es $\geq 10/t$	Tester
utilización de recursos	Utilización de CPU	¿Cuánto porcentaje de CPU es usado para realizar una tarea dada?	Tomar el porcentaje de CPU que se usa para realizar una tarea	$X = ? R/n$ N=número de pruebas	$0 \leq X \leq 1$ El más cercano a 0 es el mejor. Donde el peor caso es $\geq 15t$ .	Tester
	Utilización de la memoria	¿Cuánto espacio de memoria es usado para realizar una tarea dada?	Medir la cantidad de espacios de memoria que realmente es usado para realizar una tarea.	$X = ? R/n$ N=número de pruebas	$0 \leq X \leq 15$ El más cercano a 0 es el mejor.	Tester

**Tabla 12** Matriz de calidad externa  
Fuente: (Amparo, 2014)



<b>CRITERIOS</b>	<b>DESCRIPCIÓN</b>
<i>Subcaracterística</i>	Nombre Subcaracterística
<i>Métrica</i>	Nombre de la métrica
<i>Fórmula</i>	Fórmula de la métrica
<i>Valor deseado</i>	Umbrales de medida
<i>Valor obtenido</i>	Valor que se obtiene al momento de aplicar la formula
<i>Ponderación</i>	Valor sobre 10 según lo establecido en la tabla 9
<i>Valor total</i>	Promedio de los valores obtenidos de las métricas acuerdo a la escala de medición

**Tabla 13** Criterios de la matriz de calidad  
**Fuente:** (Amparo, 2014)



Subcaracterísticas	Métrica	Fórmula por prueba	Fórmula por todas las pruebas	Valor deseado	Valor Obtenido (R)	Ponderación (/10)	Valor Parcial Total (/10)
Eficiencia	Tiempo de respuesta	$X = B - A$ A=Tiempo de envío de petición B= Tiempo en recibir la primera respuesta	$X = \sum R/n$ n=Número de pruebas	$0 \leq X \leq 1$ El más cercano a 1 es el mejor.	R= n= X=		
	Tiempo de espera	$X = B - A$ A=Tiempo cuando se inicia un trabajo B=Tiempo en completar el trabajo	$X = \sum R/n$ n=Número de pruebas	$0 \leq X \leq 1$ El más cercano a 1 es el mejor.	R= n= X=		
	Rendimiento	$X = A/T$ A=Número de tareas completadas T=Intervalo de tiempo Donde: $T > 0$	$X = \sum R/n$ n=Número de pruebas	$0 \leq X \leq 1$ El más cercano a 1 es el mejor.	R= n= X=		
	Utilización de CPU	$X = A$ A= cantidad de CPU que es usado para realizar una tarea	$X = \sum R/n$ n=Número de pruebas	$X = A/T$ El más lejano a 0/t es el mejor	R= n= X=		
	Utilización de la memoria	$X = A$ A= Cantidad de memoria que es usado para realizar una tarea.	$X = \sum R/n$ n=Número de pruebas	$0 \leq X \leq 15$ El más cercano a 0 es el mejor.	R= n= X=		

Tabla 14 Matriz de Calidad Externa  
Fuente: (Amparo, 2014)



### 5.3 Desarrollar una aplicación móvil haciendo uso de los frameworks seleccionados

En el presente objetivo se describe los requerimientos y procesos realizados, para la aplicación móvil ADP EAT teniendo como resultado la documentación técnica de la aplicación, desarrollado bajo el marco ágil SCRUM.

#### 5.3.1 A Domicilio Perú SAC

A DOMICILIO S.A.C es una empresa de intermediación laboral de dotación de personal fundada el 28 de febrero del 2011. Tienen cuatros años en el mercado brindando soluciones externalizadas pensadas de acuerdo a las necesidades de cada uno de nuestros clientes y actualmente contamos con clientes tales como SENATI, SENCICO, CONADIS, Gobierno Regional de Lambayeque, PODER JUDICIAL, ELECTRO ORIENTE, entre otras importantes instituciones.

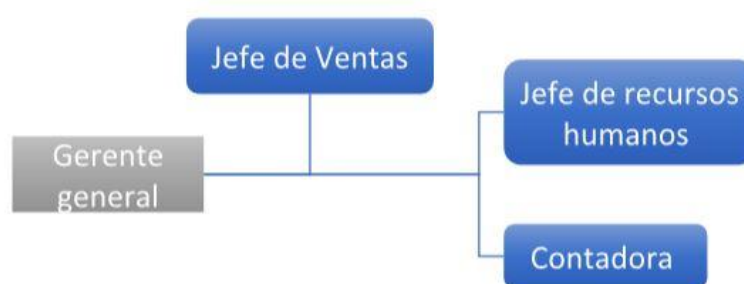
##### Visión

Destacarnos como empresa de referencia en especialista en actividades secundarias.

##### Misión

Brindar soluciones de manera oportuna, responsable y eficiente en especialistas en actividades secundarias, con alto grado de responsabilidad social y cuidado del medio ambiente.

##### Organigrama



**Ilustración 11** Organigrama de A domicilio Perú SAC  
**Fuente:** Elaboración propia



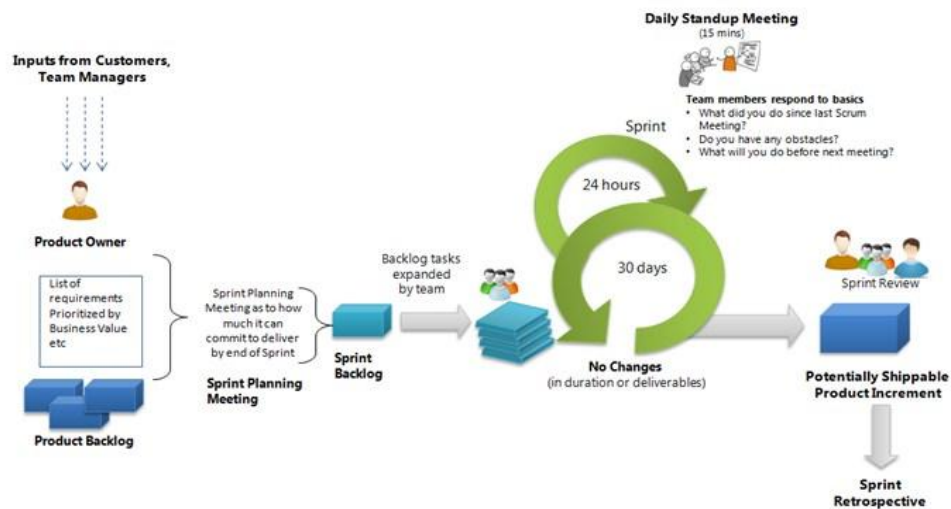
### 5.3.2 Desarrollo de la aplicación móvil ADP EAT

Para el desarrollo de la aplicación móvil ADP EAT se utilizó la metodología ágil SCRUM, el cual permitió que el desarrollo de la aplicación se vaya ajustando a las necesidades del cliente.

El modelo consta de 3 fases:

- **Fase Pregame:** el equipo se dedica a la planificación y definición del diseño de la arquitectura de la aplicación móvil.
- **Fase Development o Game:** el equipo se encarga del desarrollo de cada Sprint planificado.
- **Fase PostGame:** equipo se encarga de revisar y evaluar los resultados de cada Sprint.

#### Agile Scrum Methodology



**Ilustración 12 Metodología SRUM**  
**Fuente:**(Prateek Aggarwal, n.d.)

El proceso completo de la aplicación del modelo para desarrollar la “aplicación móvil ADP EAT” se detalla a continuación.

Para el desarrollo de la aplicación móvil, se identificó los integrantes y roles que forman parte del equipo ágil, teniendo la siguiente Tabla 15:





ROL	RESPONSABLE	DESCRIPCIÓN
<b>Product Owner</b>	Ing. Tuesta Monteza Víctor	Dueño del negocio
<b>Scrum master y development team</b>	Guevara Zavaleta Noheli Elizabeth	Gestor y desarrollador del proyecto

**Tabla 15: Roles de SCRUM**  
**Fuente: Elaboración propia**

### 5.3.3 Planificación para el desarrollo de la aplicación móvil

La planificación del desarrollo de la aplicación móvil, estimación de duración de cada sprint y el Product Backlog, se determinó en base a la lista de requerimientos funcionales y no funcionales identificados, con el dueño del negocio.

#### Requerimientos funcionales de la aplicación móvil.

- La aplicación debe gestionar la información propia de los restaurantes.
- El sistema debe confirmar el ingreso de información.
- El sistema debe registrar usuario (restaurante)
- La interfaz restaurante debe permitir registrar datos del restaurante, menú, precios.
- La aplicación restaurante debe listar los platillos.

#### Requisitos no funcionales:

- La interfaz debe llevar los colores de la empresa.
- La aplicación móvil será desarrollada para el SO Android.
- Se utilizará un emulador para testear la app móvil.



## DOCUMENTACIÓN DE REQUISITOS

NOMBRE DEL PROYECTO	SIGLAS DEL PROYECTO
Desarrollo de una Aplicación Móvil para Restaurante.	"DAMR"

**NECESIDAD DEL NEGOCIO U OPORTUNIDAD A APROVECHAR:** *DESCRIBIR LAS LIMITACIONES DE LA SITUACIÓN ACTUAL Y LAS RAZONES POR LAS CUÁLES SE EMPRENDE EL PROYECTO.*

Mejorar y agilizar los procesos involucrados al realizar el servicio de chat interno. Ofrecer un servicio de calidad a través de un sistema web.

**OBJETIVOS DEL NEGOCIO Y DEL PROYECTO:** *DEFINIR CON CLARIDAD LOS OBJETIVOS DEL NEGOCIO Y DEL PROYECTO PARA PERMITIR LAS TRAZABILIDAD DE ÉSTOS.*

Cumplir todos los requerimientos establecidos por la Institución Publica

**REQUISITOS FUNCIONALES:** *DESCRIBIR PROCESOS DEL NEGOCIO, INFORMACIÓN, INTERACCIÓN CON EL PRODUCTO, ETC.*

STAKEHOLDER	PRIORIDAD OTORGADA POR EL STAKEHOLDER	REQUISITOS	
		CÓDIGO	DESCRIPCIÓN
	Muy alta	RF01	La aplicación debe gestionar la información propia de los restaurantes.
	Muy alta	RF02	El sistema debe confirmar el ingreso de información.
	Muy alta	RF03	El sistema debe registrar usuario (restaurante)
	alta	RF04	La interfaz restaurante debe permitir registrar datos del restaurante, menú, precios.
	alta	RF05	La aplicación restaurante debe listar los platillos.
STAKEHOLDER	PRIORIDAD OTORGADA POR EL STAKEHOLDER	REQUISITOS NO FUNCIONALES	
		CÓDIGO	DESCRIPCIÓN
	Muy alta	RNF01	La interfaz debe llevar los colores de la empresa.
	Muy alta	RNF02	La aplicación móvil será desarrollada para el SO Android.
	Muy baja	RNF03	Se utilizará un emulador para testear la app móvil.



REQUISITOS DE CALIDAD: <i>DESCRIBIR REQUISITOS RELATIVOS A NORMAS O ESTÁNDARES DE CALIDAD, O LA SATISFACCIÓN Y CUMPLIMIENTO DE FACTORES RELEVANTES DE CALIDAD.</i>			
STAKEHOLDER	PRIORIDAD OTORGADA POR EL STAKEHOLDER	REQUISITOS	
		CÓDIGO	DESCRIPCIÓN
Jefe de Proyecto	Muy Alto	Si	Uso de la Metodología SCRUM para el cumplimiento de las métricas del software.
CRITERIOS DE ACEPTACIÓN: <i>ESPECIFICACIONES O REQUISITOS DE RENDIMIENTO, FUNCIONALIDAD, ETC., QUE DEBEN CUMPLIRSE ANTES DE ACEPTAR EL PROYECTO.</i>			
CONCEPTOS	CRITERIOS DE ACEPTACIÓN		
1. TÉCNICOS	Los prototipos del diseño de interfaces deben estar realizadas bajo la herramienta Balsamiq Mockups.		
2. DE CALIDAD	El diseño y desarrollo de la aplicación se va a trabajar con la metodología SCRUM.		
3. ADMINISTRATIVOS			
4. COMERCIALES			
5. SOCIALES			
6. OTROS			
REGLAS DEL NEGOCIO: <i>REGLAS PRINCIPALES QUE FIJAN LOS PRINCIPIOS GUÍAS DE LA ORGANIZACIÓN.</i>			
IMPACTOS EN OTRAS ÁREAS ORGANIZACIONALES			
SUPUESTOS RELATIVOS A REQUISITOS			
RESTRICCIONES RELATIVAS A REQUISITOS			

En base a los requerimientos definidos, se elaboró las siguientes historias de usuario e historias técnicas:



### 5.3.4 Historias de usuario

Las historias de usuarios fueron identificadas con un código único, asignándole un nombre para su identificación, una breve descripción, y una prioridad realizada por el dueño del negocio, indicando los siguientes valores:

Valor 1: HU que tiene menor prioridad

Valor 5: HU a la de mayor prioridad

Como se muestra en la siguiente tabla 3, adjuntado el resto de historias de usuarios en el anexo 2.

<b>Historia de usuario:</b> Acceso a la aplicación		<b>ID:</b> HU1
<b>Prioridad:</b> 5		
<b>Descripción:</b>	Permite ingresar a la aplicación mediante un nombre de usuario y contraseña	
<b>Validación:</b>	Validar que el usuario este registrado en la base de datos	

**Tabla 16** formato de historia de usuario  
**Fuente:** (Hidalgo Orozco & Casigña Parra, 2015)

Las historias técnicas están basadas en los requisitos de la aplicación móvil. Una historia técnica se le asignó nombre, identificador, descripción y prioridad, como se muestra en la siguiente tabla 4, adjuntado el resto de historias técnicas en el anexo 3.

<b>HT:</b> Diseño de base de datos		<b>ID:</b> HT1
<b>Prioridad:</b> 4		
<b>Descripción:</b>	Definir el modelado de la base de datos con la que se desarrollara el aplicativo móvil.	

**Tabla 17** Formato de historia técnica  
**Fuente:** (Hidalgo Orozco & Casigña Parra, 2015)

El siguiente paso en la planificación fue la elaboración del Product Backlog.



### 5.3.5 Product Backlog

La elaboración del Product Backlog, fue resultado de organizar, en una sola lista todas las historias de usuario, en base a la prioridad, la Tabla 18 muestra el Product Backlog generado:

Orden	ID HU HT	Prioridad	Nombre
1	HT2	5	Diseño de la arquitectura de la aplicación
2	HT1	5	Diseño de base de datos
3	HT3	5	Servidores
4	HT4	5	Selección de herramientas
5	HU1	5	Acceso a la aplicación
6	HU3	5	Registrar platillo
7	HU4	5	Editar platillo
8	HU5	5	Eliminar platillo
9	HU6	5	Listar platillo

**Tabla 18** Product Backlog

**Fuente:** (Hidalgo Orozco & Casigña Parra, 2015)

En base al Product Backlog y una estimación en días del desarrollo de cada historia, se ha planificado las fechas de entregables de cada sprint, la estimación del primer sprint se presenta en la siguiente Tabla 19.

SPRINT 1	ID HU HT	Nombre	Estimación en días	Febrero
	HT1	Diseño de la arquitectura de la aplicación	3	6 – 15
	HT2	Diseño de base de datos	2	
	HT3	Servidores	3	

**Tabla 19** Planificación de la aplicación móvil

**Fuente:** (Hidalgo Orozco & Casigña Parra, 2015)

Concluyendo con la especificación de requerimientos, elaboración del Product Backlog y planificación de los Sprints, se procedió a la



ejecución del primer sprint, teniendo como evento necesario una reunión denominada Sprint Planning Meeting que se detalla a continuación:

### 5.3.6 Sprint Planning Meeting

Es una reunión al inicio de cada sprint; donde se definieron las tareas que se llevarían a cabo para desarrollar las historias de usuario que estaban involucradas en esta iteración.

Teniendo como resultado el Sprint Backlog, la Tabla 20 y 21, describe los dos primeros Sprints.

SPRINT	Historias	Nombre historia	Tareas Principales	Estimación
SPRINT 1	HT1	Diseño de la arquitectura de la aplicación	Selección de las tecnologías de comunicación	4
			Selección de SO.	4
			Selección de software	2
	HT2	Diseño de base de datos	Elaborar diagrama de entidad relación	2
			Diseño lógico	2
			Diseño físico	2
			Diccionario de base de datos	2
	HT3	Servidores	Instalación de servidor	2
			Configuraciones de servicios.	3
Instalación de entornos de desarrollo.			3	

**Tabla 20** Sprint BackLog - Sprint 1  
**Fuente** :(Hidalgo Orozco & Casigña Parra, 2015)

SPRINT	Historia	Nombre HU	Tareas Principales	Estimación
SPRINT 2	HT4	Selección de herramientas	Instalación del entorno de desarrollo para JQuery Mobile	3
			Instalación del entorno de desarrollo para Apache cordova	3
			Instalación de JQuery Mobile y Apache cordova	6
			Seleccionar versión de Android	3
	HU1		Creación de formulario de logueo	4



	Acceso a la app	Creación de la interface	2
		Validación de campos	4

**Tabla 21** Sprint Backlog - Sprint 2  
**Fuente:** (Hidalgo Orozco & Casigña Parra, 2015)

SPRINT	Historia	Nombre HU	Tareas Principales	Estimación
SPRINT 3	HU3	Registrar platillo	Crear interfaz de usuario	5
			Crear formulario	4
			Validar campos	5
	HU4	Editar platillo	Crear interfaz de usuario	5
			Crear formulario	4
			Validar campos	3
	HU5	Eliminar platillo	Crear interfaz de usuario	4
			Validar id en la base de datos	4
			Actualizar base de datos	2
	HU6	Listar platillo	Crear interfaz de usuario	4
			verificar registros en la bd	3

**Tabla 22** Sprint Backlog - Sprint 3  
**Fuente:** (Hidalgo Orozco & Casigña Parra, 2015)

### 5.3.7 SPRINT 01

Se procede a la ejecución del sprint 01 que inició el 1 de febrero del 2017 y finalizó el 15 de febrero, la característica principal de este sprint fue que se desarrolló y definió la arquitectura con la que funcionaria la app.

#### Arquitectura del aplicativo móvil

Para interactuar con la base de datos MySQL, se crear un web service, el cual es conseguir la solicitud del cliente, interactuar con la base de datos y, finalmente, dar la respuesta al cliente. El web service realiza las siguientes actividades:

- Acepta solicitudes en los métodos / POST GET
- Interactuar con la base de datos mediante la inserción/buscar datos.
- Por último, dará respuesta en formato JSON.



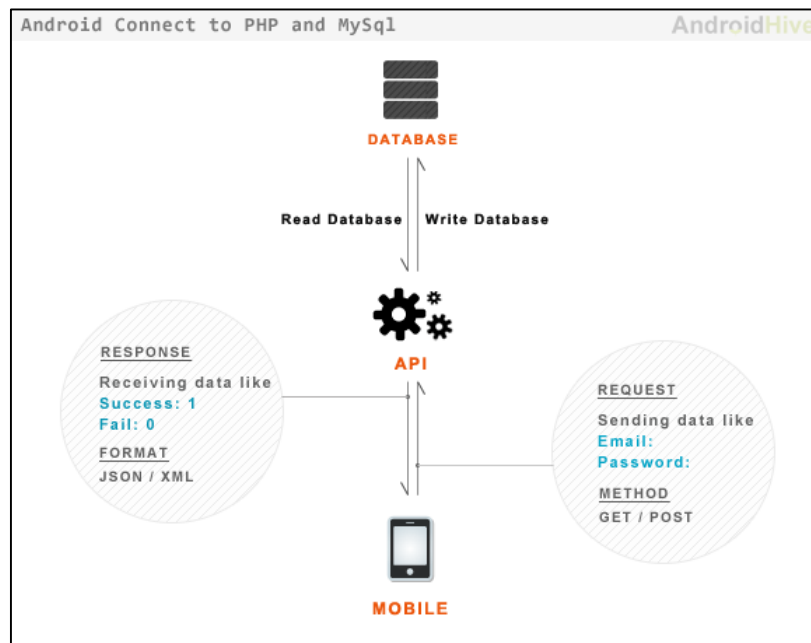


Ilustración 13 arquitectura del aplicativo  
Fuente: (Ravi Tamada, n.d.)

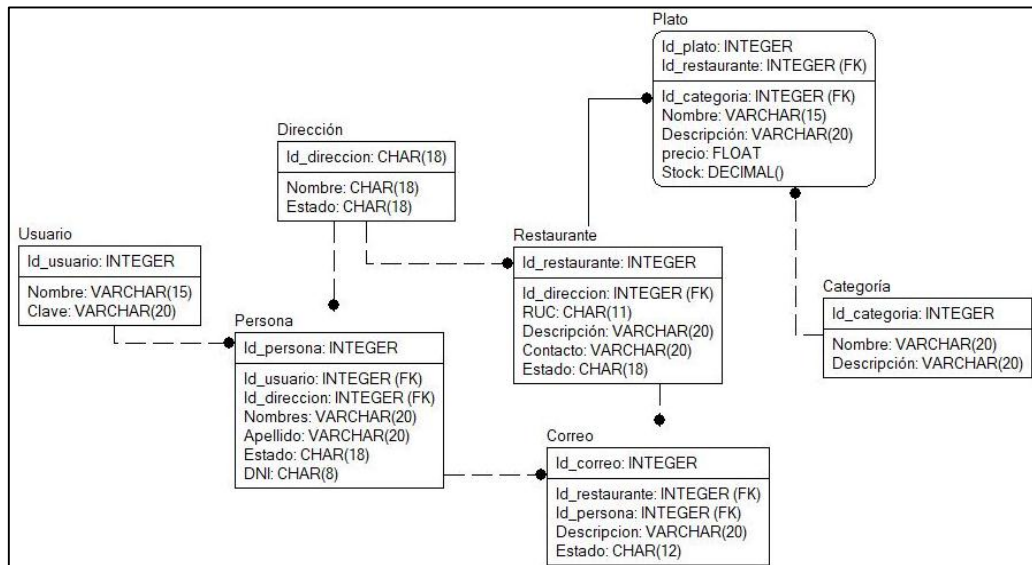
### Diseño de base de datos

A domicilio Perú SAC busca integrar a sus servicios, la gestión de pedidos de comida a domicilio, por lo que al diseño se agrega los restaurantes, comensales y repartidores. Siendo el caso, para este trabajo se consideró desarrollar una aplicación móvil para el módulo de restaurantes, que permita gestionar la información de platillos. Cada restaurante debe contar con un usuario y clave para tener acceso a la aplicación, permitiendo ingresar información de platillos, como, nombre, descripción, precio, stock y su categoría. También tiene la opción de editar y eliminar registro de platillos, según sea el caso.

En base a lo anunciado, se elaboró el siguiente modelo de base de datos relacional.







**Ilustración 14** diseño de base de datos  
**Fuente :** elaboración propia

### Diccionario de base de datos

Luego de crear el modelo de base de datos, generamos el diccionario de datos, como lo muestra la siguiente tabla 15.

Nombre tabla	Atributo	Tipo dato	Descripción	PK	FK
Restaurante	Id_restaurante	INTEGER	Guarda el código para cada restaurante.	Yes	No
	RUC	CHAR(12)	Guarda el número de RUC del restaurante, el cual tiene 11 dígitos.	No	No
	Descripción	VARCHAR(15)	Guarda la descripción de restaurante.	No	No
	Estado	CHAR(18)	Guarda el estado del restaurante	No	No
	Contacto	VARCHAR(20)	Guarda el nombre del representante de restaurante.	No	No
	Id_direccion	INTEGER	Guarda el código de dirección.	No	Yes
Categoría	Id_categoria	INTEGER	Guarda un id para cada categoría.	Yes	No
	Nombre	VARCHAR(20)	Guarda el nombre de la categoría de comida.	No	No
	Descripción	VARCHAR(25)	Guarda la descripción adicional de la categoría.	No	No



Usuario	Id_usuario	INTEGER	Guarda un código único por cada usuario.	Yes	No
	Nombre	VARCHAR(15)	Guarda el nombre de usuario para el logueo.	No	No
	Clave	VARCHAR(20)	Guarda la clave para cada usuario, que tenga acceso al sistema.	No	No
Plato	Id_plato	INTEGER	Código de plato.	Yes	No
	Nombre	VARCHAR(15)	Guardar el nombre del plato	No	No
	Descripción	VARCHAR(25)	Guarda una breve descripción del plato.	No	No
	precio	DECIMAL	Guarda el precio de cada plato.	No	No
	Id_categoria	INTEGER	Guarda un id para cada categoría.	No	Yes
	Stock	INTEGER	Guarda el stock de cada plato	No	No
Dirección	Id_direccion	INTEGER	Guarda el id para cada dirección.	No	No
	Descripcion	VARCHAR(20)	Guardar la dirección de los usuarios (restaurante, repartidor y comensal).	No	No
	Estado	CHAR(10)	Guarda el estado de la dirección, si está disponible o no.	No	No
Persona	Id_persona	INTEGER	Guarda un código para cada usuario.	Yes	No
	Nombres	VARCHAR(15)	Guarda el nombre de cada usuario.	No	No
	Estado	CHAR(5)	Guarda el estado de cada usuario, si está disponible o no.	No	No
	Id_direccion	INTEGER	Guarda el id para cada dirección.	No	Yes
	DNI	CHAR(8)	Guarda el número de DNI de cada usuario, el campo solo admite 8 dígitos.	No	No
	Id_usuario	INTEGER	Guarda un código único por cada usuario.	No	Yes
Correo	Apellido	VARCHAR(15)	Guarda el apellido de cada usuario.	No	No
	Id_correo	INTEGER	Guarda un id de correo.	Yes	No



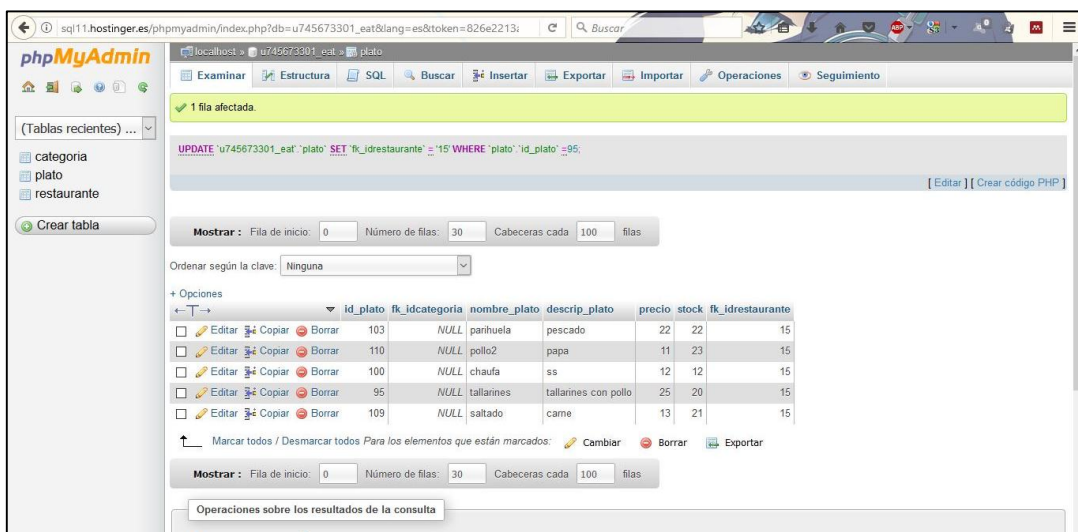
	Descripcion	VARCHAR2(20)	Guarda una descripción adicional acerca del correo.	No	No
	Estado	CHAR(12)	Guarda un estado para el correo.	No	No
	Id_persona	INTEGER	Guarda un código para cada usuario.	No	Yes
	Id_restaurante	INTEGER	Guarda el código para cada restaurante.	No	No

**Tabla 23** Diccionario de datos  
Fuente: Elaboración propia

## Servidores

### Servidor de Base de Datos

Se definió el esquema de la base de datos, y se procedió a crear una cuenta de hosting, para alojamiento del servidor de base datos siendo MySQL y el gestor de administración PhpMyAdmin.

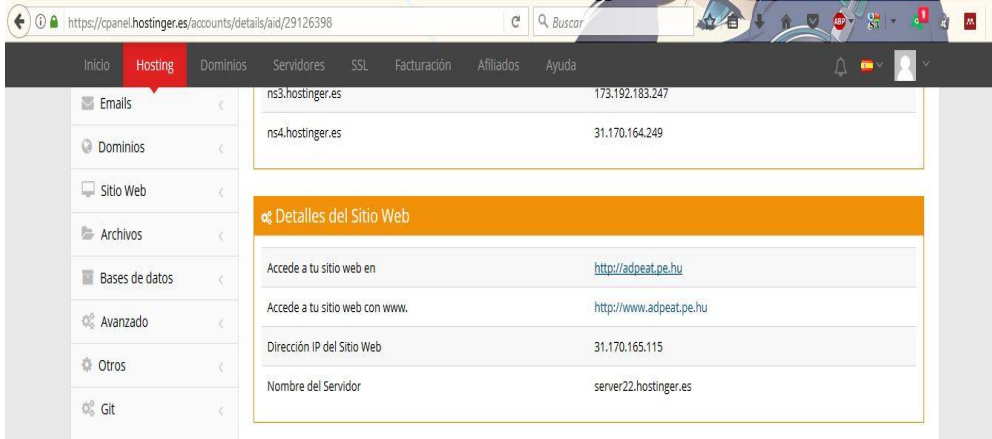


**Ilustración 15** base de datos MySQL  
Fuente: elaboración propia



### Servidor de archivos

Para alojamiento de servicio web se configuro lo siguiente:



**Ilustración 16** servidor de archivos  
Fuente: elaboración propia

### Proyecto web service:

La clase de “conexiónDAO.php” permite la conexión con la base de datos que está alojado en un hosting:

```

ConexionDao.php
Source History
<?php
header("Access-Control-Allow-Origin: *");
class ConexionDao {
    protected $dblink = null;
    public $host = "mysql.hostinger.es";
    public $username = "u745673301_eliza";
    public $password = "123456789";
    public $databasename = "u745673301_eat";
    public $charset = 'utf8';
    function __construct() {
        $opt = [
            PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
            PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
            PDO::ATTR_EMULATE_PREPARES => false,
        ];
        $this->dblink = new PDO("mysql:host=$this->host;dbname=$this->databasename;charset=$this->charset",
            $this->username, $this->password, $opt);
        $this->dblink->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $this->dblink->setAttribute(PDO::ATTR_CASE, PDO::CASE_LOWER);
        $this->dblink->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
    }
}
    
```

**Ilustración 17** Conexión BD  
Fuente: elaboración propia



La clase “Métodos.php”, contiene las funciones de logueo, insertar, editar, eliminar y actualizar:

```

1  <?php
2  header("Access-Control-Allow-Origin: *");
3  require_once '../Dao/ConexionDao.php';
4
5  class Metodos extends ConexionDao {
6
7      function LogearseApp($usuario, $password) {
8          $rec = null;
9          try {
10             $query = "select * from restaurante where usuario='$usuario' and clave='$password'";
11             $stm = $this->dblink->prepare($query);
12             $stm->execute();
13             $rec = $stm->fetch(PDO::FETCH_ASSOC);
14         } catch (Exception $ex) {
15             $rec = null;
16         }
17         return $rec;
18     }
19
20     function ListarPlatos() {
21         $rec = null;
22         try {
23             $query = "select * from plato";
24             $stm = $this->dblink->prepare($query);
25             $stm->execute();
26             $rec = $stm->fetchAll(PDO::FETCH_ASSOC);
27         } catch (Exception $e) {
28             $rec = null;
29         }
30         return $rec;
31     }
32 }
    
```

**Ilustración 18** clase métodos.php  
Fuente: elaboración propia

### 5.3.8 SPRINT 2: Herramientas

#### - Instalación de apache cordova

Instalar la CLI de Cordova

La herramienta de línea de comandos de Cordova se distribuye como un paquete de npm en un formato listo para usar.

Para instalar la herramienta de línea de comandos de cordova, se realizó los siguientes pasos:

Descargar e instale Node.js. Instale el módulo cordova utilizando el manejador de paquetes de Node.js npm. El módulo cordova será descargado automáticamente por npm.

```
C:\>npm install -g cordova
```

#### Crear aplicación

Ejecutar en cmd, en la carpeta de proyecto el siguiente comando:

```
$ cordova create APP com.example.hello APPEAT
```



## Añadir plataforma Android

```
$ cordova platform add Android
```

### - Instalación de JQuery Mobile

Para trabajar con este framework, se debe incluir las librerías de JQuery Mobile en el proyecto, quedando de la siguiente manera:

```

1 <!DOCTYPE html>
2
3 <html lang="es">
4   <head>
5     <title>ADP EAT</title>
6     <meta charset="UTF-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <link rel="stylesheet" type="text/css" href="css/index.css">
9     <link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css" />
10    <script src="http://code.jquery.com/jquery-1.11.1.min.js"></script>
11    <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
12  </head>
13
14  <body>
15    <div data-role="page">
16      <div data-role="header">
17        <h1 id="textol">A DomicilioEat</h1>
18      </div>
19      <!-- /header -->
20      <div data-role="main" class="ui-content" style="margin-top: 450px;">
21        <a href="sesion.php" data-inline="true" class="ui-btn" id="iniciar" data-role="button">INICIAR SESION</a>
22        <a href="registrar3.php" data-inline="true" class="ui-btn" id="registrar" data-role="button">REGISTRARSE</a>
23      </div>
24      <!-- /content -->
25    </div>
26    <!-- /page -->
27  </body>
28 </html>
29
  
```

**Ilustración 19** librerías de JQuery Mobile  
Fuente: elaboración propia

### Seleccionar versión de Android

De acuerdo a informes realizados, Marshmallow ha podido finalmente superar a KitKat y a Lollipop para pasar a convertirse en la versión de Android más usada en el mundo estando el 26.3% de dispositivos.

Sin embargo, si juntamos Lollipop 5.0 y Lollipop 5.1 esta saldría como ganadora, siendo la versión que, si bien ya es bastante estable y cuenta con varias mejoras sustanciales, ya tiene más de tres años de existencia. (Melly, 2016)

Para la presente investigación se consideró utilizar la última versión disponible, siendo el caso de Nougat versión 7.0.



Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.2%
4.1.x	Jelly Bean	16	4.5%
4.2.x		17	6.4%
4.3		18	1.9%
4.4	KitKat	19	24.0%
5.0	Lollipop	21	10.8%
5.1		22	23.2%
6.0	Marshmallow	23	26.3%
7.0	Nougat	24	0.4%

**Ilustración 20** Versión de SO Android  
Fuente (Melly, 2016)

**Acceso a la app**

The image shows a mobile application interface on the left and its underlying code on the right. The interface features a 'REGISTRARSE' button at the top, followed by a background image of a restaurant. Below the image are input fields for 'Usuario' and 'Contraseña', a link for 'HAZ OLVIDADO LA CONTRASEÑA?', and an 'ACEPTAR' button. The code on the right is split into two panels: HTML and CSS. The HTML panel shows the structure of the registration form, including a menu, a login form with input fields and a submit button, and a footer. The CSS panel shows styles for the menu and body, including background color, display, height, and font-family.

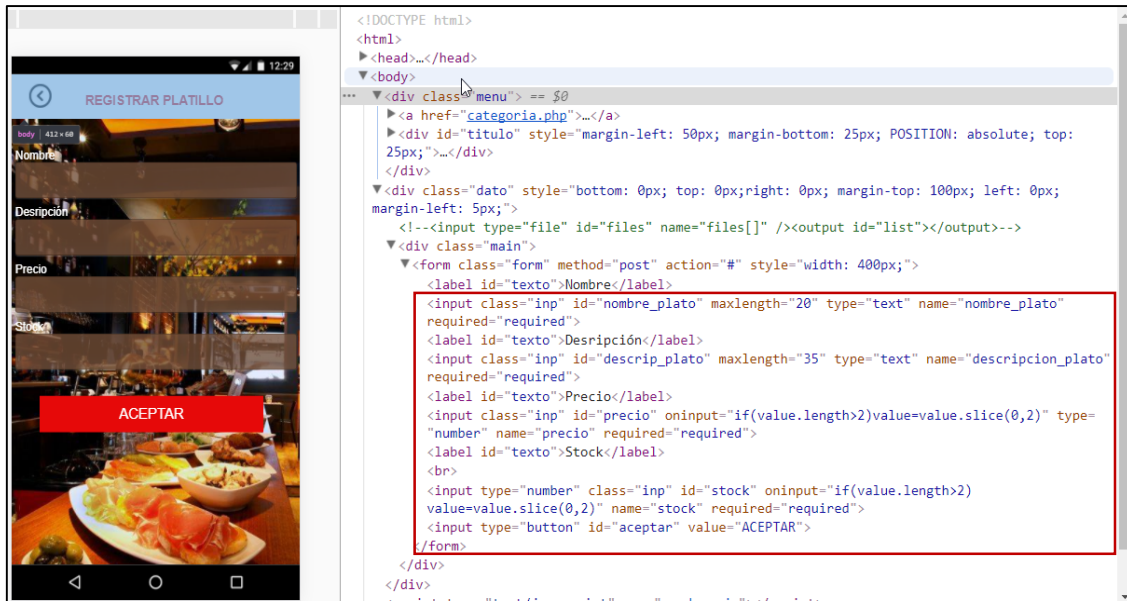
- 1.- interfaz de usuario.
- 2.- creación de formulario para logueo.
- 3.- estilos css.



### 5.3.9 SPRINT 03

#### Registrar platillo

Para cumplir con esta historia de usuario, se diseñó la siguiente interfaz que permite crear un platillo, para lo cual se creó un formulario, y se validó los campos, como lo muestra la siguiente ilustración:



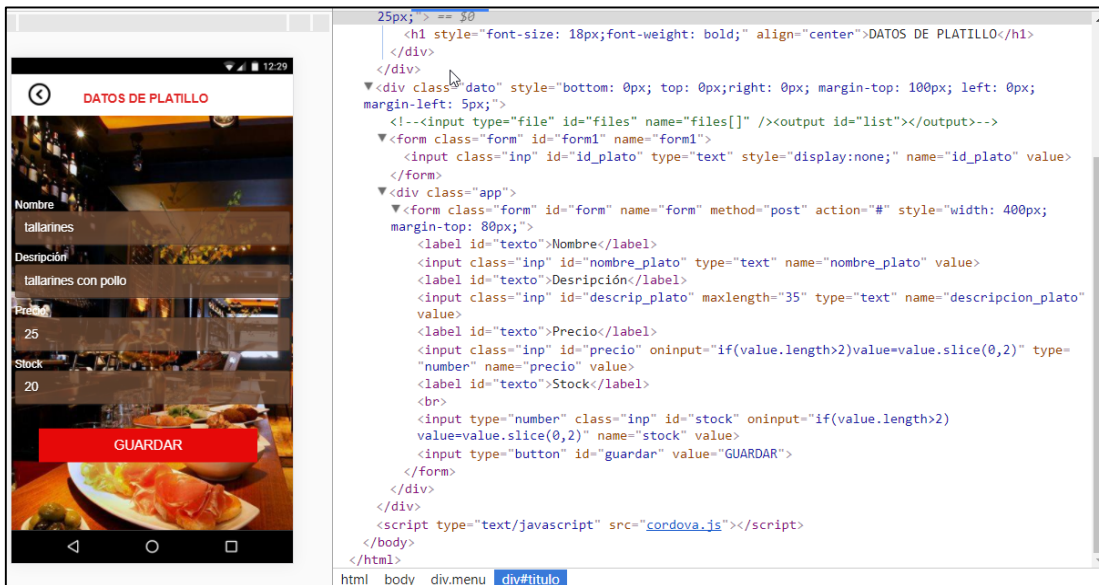
Fuente: elaboración propia

#### Editar platillo

Para implementar esta historia de usuario, se realizaron las siguientes actividades, diseño de interfaz, crear un formulario que permita mostrar e ingresar nuevos datos, se muestra en la siguiente ilustración:

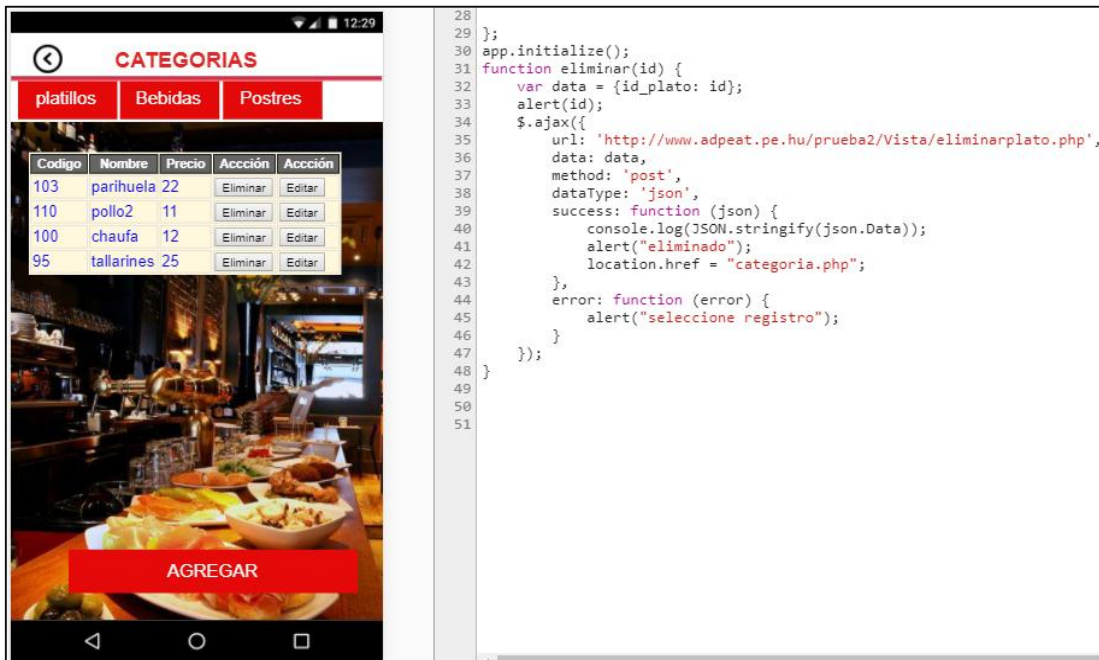






### Eliminar platillo

Para eliminar platillo, se diseñó la interfaz de usuario con colores representativos de la empresa, realizar una consulta a la base de datos para verificar el id de registro a eliminar:





instante en que finalizó. En cuanto a las mediciones correspondientes a la utilización de memoria, éstas se obtuvieron analizando el periodo de tiempo que tardó en completarse una tarea y calculando la cantidad de memoria ocupada en el instante en que la tarea se realizó. Por otro lado, para las mediciones de utilización de CPU, haciendo uso del lapso de tiempo que tardó una tarea en ser completada, se identificó el pico más elevado registrado en dicho periodo que se interpreta como el uso máximo de CPU en la ejecución de la respectiva tarea. (Sanjuán et al., 2015)

Los resultados obtenidos de la medición de calidad externa de cada aplicación móvil son los siguientes:

#### **5.4.2 Aplicación móvil desarrollada con Apache Cordova:**

##### **5.4.2.1 Métrica: Tiempo de respuesta:**

Datos obtenidos por cada tarea, haciendo uso de la formula  $X=B - A$  A=Tiempo de envío de petición B= Tiempo en recibir la primera respuesta.



**TIEMPO DE RESPUESTA**

N° pruebas	tarea 1			tarea 2			tarea 3			tarea 4			tarea 5		
	A	B	resultado	A	B	resultado	A	B	resultado	A	B	resultado	A	B	resultado
prueba 1	40.56	44.55	3.990	1.81	5.52	3.71	6.63	9.12	2.49	4.31	9.25	4.94	49.91	53.12	3.21
prueba 2	46.37	50.26	3.890	17.71	21.76	4.05	31.26	34.13	2.87	11.66	19.48	7.82	5.56	8.26	2.70
prueba 3	3.29	6.04	2.750	44.88	47.25	2.37	29.32	31.79	2.47	57.7	65.86	8.16	27.19	31.65	4.46
prueba 4	23.61	26.26	2.650	13.9	16.09	2.19	21.79	24.61	2.82	30.44	35.68	5.24	58.6	59.75	1.15
prueba 5	31.94	33.12	1.180	56.96	59.23	2.27	54.14	56.39	2.25	44.96	47.26	2.30	35.57	38.35	2.78
prueba 6	5.45	8.50	3.050	14.23	17.45	3.22	14.26	17.54	3.28	55.51	58.26	2.75	15.23	18.21	2.98
prueba 7	32.47	36.15	3.680	25.47	27.56	2.09	25.26	28.06	2.80	44.12	47.52	3.40	36.15	38.60	2.45
prueba 8	47.23	50.59	3.360	45.21	47.25	2.04	18.51	21.43	2.92	32.26	36.23	3.97	54.32	57.11	2.79
prueba 9	15.25	18.56	3.310	32.26	35.54	3.28	27.22	30.44	3.22	15.23	19.12	3.89	56.21	59.54	3.33
prueba 10	32.45	35.52	3.070	50.12	52.24	2.12	24.16	26.25	2.09	45.32	49.05	3.73	48.02	50.33	2.31
prueba 11	51.26	54.10	2.840	16.25	19.32	3.07	43.21	46.47	3.26	47.04	50.45	3.41	19.25	23.59	4.34
prueba 12	8.24	10.42	2.180	47.11	49.21	2.10	32.47	35.03	2.56	26.32	29.45	3.13	34.54	37.33	2.79
prueba 13	25.36	28.21	2.850	53.32	55.66	2.34	41.33	43.55	2.22	33.55	36.21	2.66	26.54	29.35	2.81
prueba 14	18.45	20.23	1.780	15.26	18.54	3.28	53.47	56.33	2.86	46.53	49.47	2.94	11.24	15.35	4.11
prueba 15	28.56	31.12	2.560	9.26	12.20	2.94	8.62	11.32	2.70	36.25	39.54	3.29	28.22	31.02	2.80
prueba 16	43.21	45.56	2.350	36.55	39.48	2.93	34.26	37.33	3.07	53.26	58.36	5.10	11.56	14.55	2.99
prueba 17	54.24	57.20	2.960	47.58	50.57	2.99	16.05	19.22	3.17	15.48	19.45	3.97	47.55	51.03	3.48
prueba 18	6.12	8.45	2.330	12.56	16.47	3.91	31.52	34.05	2.53	19.3	22.12	2.82	15.21	18.55	3.34
prueba 19	16.32	19.15	2.830	51.32	54.11	2.79	29.54	32.41	2.87	22.15	24.56	2.41	53.44	55.54	2.10
prueba 20	24.31	27.56	3.250	36.25	39.23	2.98	27.55	29.66	2.11	48.32	50.34	2.02	9.44	13.20	3.76
prueba 21	31.26	34.21	2.950	33.51	36.54	3.03	22.35	25.51	3.16	35.14	39.02	3.88	16.25	19.05	2.80
prueba 22	55.47	58.49	3.020	21.22	26.12	4.90	21.54	23.12	1.58	20.47	24.11	3.64	36.45	39.14	2.69
prueba 23	11.32	13.41	2.090	14.45	18.23	3.78	36.21	39.05	2.84	15.16	19.45	4.29	38.11	41.36	3.25
prueba 24	21.46	24.31	2.850	36.16	38.45	2.29	25.12	28.32	3.20	41.05	43.48	2.43	24.26	27.24	2.98
prueba 25	45.53	48.32	2.790	38.41	41.32	2.91	38.56	41.21	2.65	8.32	10.59	2.27	38.25	41.44	3.19
prueba 26	44.41	46.54	2.130	37.24	40.22	2.98	14.23	17.02	2.79	14.26	19.27	5.01	45.18	49.17	3.99
prueba 27	32.45	35.54	3.090	45.56	48.33	2.77	15.02	17.07	2.05	25.33	29.35	4.02	49.26	52.06	2.80
prueba 28	13.1	15.26	2.160	17.15	20.05	2.90	52.22	55.33	3.11	42.57	46.06	3.49	16.11	19.35	3.24
prueba 29	29.28	32.12	2.840	22.54	25.55	3.01	41.11	43.56	2.45	13.25	16.47	3.22	35.22	39.06	3.84
prueba 30	12.14	15.35	3.210	21.32	23.45	2.13	47.23	50.03	2.80	51.26	56.14	4.88	16.33	19.26	2.93

**Tabla 24** Tiempo de Respuesta

Fuente: elaboración propia



Luego se obtuvo una sumatoria de todas las tareas por cada prueba:

TIEMPO DE RESPUESTA						
	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5	Resultados
Prueba 1	3.99	3.71	2.49	4.94	3.21	3.668
Prueba 2	3.89	4.05	2.87	7.82	2.7	4.266
Prueba 3	2.75	2.37	2.47	8.16	4.46	4.042
Prueba 4	2.65	2.19	2.25	5.24	1.15	2.696
Prueba 5	1.18	2.27	2.25	2.3	2.78	2.156
Prueba 6	3.05	3.22	3.28	2.75	2.98	3.056
Prueba 7	3.68	2.09	2.8	3.4	2.45	2.884
Prueba 8	3.36	2.04	2.92	3.97	2.79	3.016
Prueba 9	3.31	3.28	3.22	3.89	3.33	3.406
Prueba 10	3.07	2.12	2.09	3.73	2.31	2.664
Prueba 11	2.84	3.07	3.26	3.41	4.34	3.384
Prueba 12	2.18	2.1	2.56	3.13	2.79	2.552
Prueba 13	2.85	2.34	2.22	2.66	2.81	2.576
Prueba 14	1.78	3.28	2.86	2.94	4.11	2.994
Prueba 15	2.56	2.94	2.7	3.29	2.8	2.858
Prueba 16	2.35	2.93	3.07	5.1	2.99	3.288
Prueba 17	2.96	2.99	3.17	3.97	3.48	3.314
Prueba 18	2.33	3.91	2.53	2.82	3.34	2.986
Prueba 19	2.83	2.79	2.87	2.41	2.1	2.6
Prueba 20	3.25	2.98	2.11	2.02	3.76	2.824
Prueba 21	2.95	3.03	3.16	3.88	2.8	3.164
Prueba 22	3.02	4.9	1.58	3.64	2.69	3.166
Prueba 23	2.09	3.78	2.84	4.29	3.25	3.25
Prueba 24	2.85	2.29	3.2	2.43	2.98	2.75
Prueba 25	2.79	2.91	2.65	2.27	3.19	2.762
Prueba 26	2.13	2.98	2.79	5.01	3.99	3.38
Prueba 27	3.09	2.77	2.05	4.02	2.8	2.946
Prueba 28	2.16	2.9	3.11	3.49	3.24	2.98
Prueba 29	2.84	3.01	2.45	3.22	3.84	3.072
Prueba 30	3.21	2.13	2.8	4.88	2.93	3.19

**Tabla 25** Tiempo de respuesta – sumatoria

**Fuente:** elaboración propia

Por último, se calculó la sumatoria de tiempo de respuesta de la tarea para todas las pruebas, siendo 91.89 que viene hacer la variable  $\sum R$  en la matriz de calidad.



### 5.4.2.2 Métrica: Tiempo de espera:

Datos obtenidos por cada tarea, haciendo uso de la fórmula:  $X = B - A$   
 $A$ =Tiempo cuando se inicia un trabajo     $B$ =Tiempo en completar el trabajo.

N° pruebas	TIEMPO DE ESPERA														
	tarea 1			tarea 2			tarea 3			tarea 4			tarea 5		
	A	B	resultado	A	B	resultado	A	B	resultado	A	B	resultado	A	B	resultado
Prueba 1	30.45	45.58	15.130	27.51	32.12	4.61	59.1	78.58	19.48	4.11	11.97	7.86	30.78	50.40	19.62
Prueba 2	20.68	36.15	15.470	17.71	21.93	4.22	16.62	35.34	18.72	11.72	22.34	10.62	29.04	46.93	17.89
Prueba 3	40.19	66.04	25.850	44.88	47.76	2.88	29.32	65.62	36.30	57.7	66.45	8.75	1.19	33.94	32.75
Prueba 4	49.83	85.40	35.570	13.84	16.75	2.91	14.95	55.19	40.24	30.48	40.95	10.47	31.86	60.68	28.82
Prueba 5	55.6	95.93	40.330	56.41	58.89	2.48	57.65	98.15	40.50	44.75	55.86	11.11	55.33	75.47	20.14
Prueba 6	35	65.45	30.450	41.23	44.25	3.02	45.21	79.26	34.05	59.76	80.20	20.44	26.31	46.36	20.05
Prueba 7	26.21	46.35	20.140	15.26	19.15	3.89	51.2	85.32	34.12	62.32	81.23	18.91	21.25	45.56	24.31
Prueba 8	45.26	66.21	20.950	48.23	51.02	2.79	15.26	45.12	29.86	15.2	36.38	21.18	36.54	56.35	19.81
Prueba 9	15.21	35.23	20.020	35.26	38.14	2.88	30.26	62.23	31.97	56.32	71.20	14.88	16.25	37.56	21.31
Prueba 10	32.25	66.23	33.980	48.26	50.48	2.22	31.25	56.16	24.91	14.15	26.11	11.96	14.21	35.26	21.05
Prueba 11	15.23	37.48	22.250	54.22	59.26	5.04	12.03	36.54	24.51	21.32	33.16	11.84	26.32	46.35	20.03
Prueba 12	25.36	48.25	22.890	26.58	29.26	2.68	15.26	35.27	20.01	14.41	23.10	8.69	34.32	56.21	21.89
Prueba 13	45.23	80.15	34.920	32.15	35.32	3.17	47.12	75.02	27.90	25.32	35.12	9.80	45.32	63.45	18.13
Prueba 14	10.41	28.35	17.940	33.26	36.52	3.26	21.04	48.15	27.11	51.02	59.23	8.21	16.35	36.05	19.70
Prueba 15	19.23	41.12	21.890	42.35	45.11	2.76	41.47	62.32	20.85	11.31	22.15	10.84	41.25	64.51	23.26
Prueba 16	26.32	48.32	22.000	50.14	53.26	3.12	15.21	35.16	19.95	16.48	24.23	7.75	11.26	35.24	23.98
Prueba 17	41.32	62.23	20.910	52.26	55.03	2.77	21.42	42.15	20.73	21.26	28.59	7.33	8.45	29.31	20.86
Prueba 18	25.26	45.21	19.950	45.32	48.58	3.26	32.14	52.12	19.98	41.11	48.54	7.43	12.3	38.21	25.91
Prueba 19	36.54	66.56	30.020	21.02	25.48	4.46	26.3	46.23	19.93	5.12	11.32	6.20	26.34	46.53	20.19
Prueba 20	21.51	41.03	19.520	45.43	48.54	3.11	12.25	36.15	23.90	34.1	40.23	6.13	11.26	33.46	22.20
Prueba 21	22.35	43.15	20.800	25.12	28.15	3.03	24.26	46.13	21.87	12.25	19.26	7.01	45.25	65.47	20.22
Prueba 22	17.52	37.45	19.930	32.26	34.54	2.28	11.32	42.3	30.98	37.52	45.32	7.80	35.1	56.36	21.26
Prueba 23	26.53	48.12	21.590	47.11	50.12	3.01	23.26	45.47	22.21	16.25	23.56	7.31	6.54	26.18	19.64
Prueba 24	51.03	75.26	24.230	21.26	26.34	5.08	33.57	59.21	25.64	36.12	42.15	6.03	18.32	39.56	21.24
Prueba 25	24.22	46.25	22.030	8.41	11.03	2.62	13.41	42.36	28.95	42.25	49.26	7.01	25.36	45.32	19.96
Prueba 26	15.54	35.02	19.480	12.32	15.12	2.80	27.58	56.21	28.63	31.26	38.15	6.89	21.16	45.27	24.11
Prueba 27	26.36	46.15	19.790	45.02	47.32	2.30	15.47	36.12	20.65	16.32	22.51	6.19	28.31	48.06	19.75
Prueba 28	21.17	39.26	18.090	26.14	29.54	3.40	26.35	53.12	26.77	41.3	49.36	8.06	31.05	55.26	24.21
Prueba 29	54.01	75.52	21.510	25.03	29.55	4.52	41.47	60.15	18.68	36.11	43.15	7.04	27.16	45.39	18.23
Prueba 30	33.06	55.02	21.960	32.15	34.25	2.10	26.18	61.32	35.14	28.34	35.12	6.78	15.29	35.28	19.99

**Tabla 26** Tiempo de espera  
**Fuente:** Elaboración propia



Luego se obtuvo una sumatoria de todas las tareas por cada prueba:

**TIEMPO DE ESPERA**

N° Prueba	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5	Resultados
Prueba 1	15.13	4.61	19.48	7.86	19.62	13.34
Prueba 2	15.47	4.22	18.72	10.62	17.89	13.384
Prueba 3	25.85	2.88	36.3	8.75	32.75	21.306
Prueba 4	35.57	2.91	40.24	10.47	28.82	23.602
Prueba 5	40.33	2.48	40.5	11.11	20.14	22.912
Prueba 6	30.45	3.02	34.05	20.44	20.05	21.602
Prueba 7	20.14	3.89	34.12	18.91	24.31	20.274
Prueba 8	20.95	2.79	29.86	21.18	19.81	18.918
Prueba 9	20.02	2.88	31.97	14.88	21.31	18.212
Prueba 10	33.98	2.22	24.91	11.96	21.05	18.824
Prueba 11	22.25	5.04	24.51	11.84	20.03	16.734
Prueba 12	22.89	2.68	20.01	8.69	21.89	15.232
Prueba 13	34.92	3.17	27.9	9.8	18.13	18.784
Prueba 14	17.94	3.26	27.11	8.21	19.7	15.244
Prueba 15	21.89	2.76	20.85	10.84	23.26	15.92
Prueba 16	22	3.12	19.95	7.75	23.98	15.36
Prueba 17	20.91	2.77	20.73	7.33	20.86	14.52
Prueba 18	19.95	3.26	19.98	7.43	25.91	15.306
Prueba 19	30.02	4.46	19.93	6.2	20.19	16.16
Prueba 20	19.52	3.11	23.9	6.13	22.2	14.972
Prueba 21	20.8	3.03	21.87	7.01	20.22	14.586
Prueba 22	19.93	2.28	30.98	7.8	21.26	16.45
Prueba 23	21.59	3.01	22.21	7.31	19.64	14.752
Prueba 24	24.23	5.08	25.64	6.03	21.24	16.444
Prueba 25	22.03	2.62	28.95	7.01	19.96	16.114
Prueba 26	19.48	2.8	28.63	6.89	24.11	16.382
Prueba 27	19.79	2.3	20.65	6.19	19.75	13.736
Prueba 28	18.09	3.4	26.77	8.06	24.21	16.106
Prueba 29	21.51	4.52	18.68	7.04	18.23	13.996
Prueba 30	21.96	2.1	35.14	6.78	19.99	17.194

**Tabla 27** Tiempo de espera – sumatoria

**Fuente:** Elaboración propia

Por último, se calculó la sumatoria de tiempo de respuesta de la tarea para todas las pruebas, siendo 506.366 que viene hacer la variable  $\sum R$  en la matriz de calidad.



5.4.2.3 Métrica: Rendimiento:

Datos obtenidos por cada tarea, haciendo uso de la fórmula:  $X=A/T$   
 A=Número de tareas completadas T=Intervalo de tiempo Donde:  
 $T>0$

N PRUEBAS	T	T	T	T	T
Prueba 1	15.13	4.61	19.48	7.86	19.62
Prueba 2	15.47	4.22	18.72	10.62	17.89
Prueba 3	25.85	2.88	36.30	8.75	32.75
Prueba 4	35.57	2.91	40.24	10.47	28.82
Prueba 5	40.33	2.48	40.50	11.11	20.14
Prueba 6	30.45	3.02	34.05	20.44	20.05
Prueba 7	20.14	3.89	34.12	18.91	24.31
Prueba 8	20.95	2.79	29.86	21.18	19.81
Prueba 9	20.02	2.88	31.97	14.88	21.31
Prueba 10	33.98	2.22	24.91	11.96	21.05
Prueba 11	22.25	5.04	24.51	11.84	20.03
Prueba 12	22.89	2.68	20.01	8.69	21.89
Prueba 13	34.92	3.17	27.90	9.80	18.13
Prueba 14	17.94	3.26	27.11	8.21	19.70
Prueba 15	21.89	2.76	20.85	10.84	23.26
Prueba 16	22.00	3.12	19.95	7.75	23.98
Prueba 17	20.91	2.77	20.73	7.33	20.86
Prueba 18	19.95	3.26	19.98	7.43	25.91
Prueba 19	30.02	4.46	19.93	6.20	20.19
Prueba 20	19.52	3.11	23.90	6.13	22.20
Prueba 21	20.80	3.03	21.87	7.01	20.22
Prueba 22	19.93	2.28	30.98	7.80	21.26
Prueba 23	21.59	3.01	22.21	7.31	19.64
Prueba 24	24.23	5.08	25.64	6.03	21.24
Prueba 25	22.03	2.62	28.95	7.01	19.96
Prueba 26	19.48	2.80	28.63	6.89	24.11
Prueba 27	19.79	2.30	20.65	6.19	19.75
Prueba 28	18.09	3.40	26.77	8.06	24.21
Prueba 29	21.51	4.52	18.68	7.04	18.23
Prueba 30	21.96	2.10	35.14	6.78	19.99

**Tabla 28** Tiempo de Rendimiento  
**Fuente:** Elaboración propia





Luego se obtuvo una sumatoria de todas las tareas por cada prueba:

TIEMPO DE RENDIMIENTO						
N° Prueba	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5	Resultados
Prueba 1	15.13	4.61	19.48	7.86	19.62	0.37
Prueba 2	15.47	4.22	18.72	10.62	17.89	0.37
Prueba 3	25.85	2.88	36.30	8.75	32.75	0.23
Prueba 4	35.57	2.91	40.24	10.47	28.82	0.21
Prueba 5	40.33	2.48	40.50	11.11	20.14	0.22
Prueba 6	30.45	3.02	34.05	20.44	20.05	0.23
Prueba 7	20.14	3.89	34.12	18.91	24.31	0.25
Prueba 8	20.95	2.79	29.86	21.18	19.81	0.26
Prueba 9	20.02	2.88	31.97	14.88	21.31	0.27
Prueba 10	33.98	2.22	24.91	11.96	21.05	0.27
Prueba 11	22.25	5.04	24.51	11.84	20.03	0.30
Prueba 12	22.89	2.68	20.01	8.69	21.89	0.33
Prueba 13	34.92	3.17	27.90	9.80	18.13	0.27
Prueba 14	17.94	3.26	27.11	8.21	19.70	0.33
Prueba 15	21.89	2.76	20.85	10.84	23.26	0.31
Prueba 16	22.00	3.12	19.95	7.75	23.98	0.33
Prueba 17	20.91	2.77	20.73	7.33	20.86	0.34
Prueba 18	19.95	3.26	19.98	7.43	25.91	0.33
Prueba 19	30.02	4.46	19.93	6.20	20.19	0.31
Prueba 20	19.52	3.11	23.90	6.13	22.20	0.33
Prueba 21	20.80	3.03	21.87	7.01	20.22	0.34
Prueba 22	19.93	2.28	30.98	7.80	21.26	0.30
Prueba 23	21.59	3.01	22.21	7.31	19.64	0.34
Prueba 24	24.23	5.08	25.64	6.03	21.24	0.30
Prueba 25	22.03	2.62	28.95	7.01	19.96	0.31
Prueba 26	19.48	2.80	28.63	6.89	24.11	0.31
Prueba 27	19.79	2.30	20.65	6.19	19.75	0.36
Prueba 28	18.09	3.40	26.77	8.06	24.21	0.31
Prueba 29	21.51	4.52	18.68	7.04	18.23	0.36
Prueba 30	21.96	2.10	35.14	6.78	19.99	0.29

**Tabla 29** Tiempo de rendimiento – sumatoria

**Fuente:** Elaboración propia

Por último, se calculó la sumatoria de rendimiento de la tarea para todas las pruebas, siendo 9.10 que viene hacer la variable  $\sum R$  en la matriz de calidad.



#### 5.4.2.4 Métrica: Utilización de CPU:

Datos obtenidos por cada tarea, haciendo uso de la formula  $X = \frac{A}{N}$   
 A = cantidad de CPU que es usado para realizar una tarea

UTILIZACIÓN DE CPU

	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5
N pruebas	A	A	A	A	A
Prueba 1	59.54	57.26	60.27	44.12	71.00
Prueba 2	60.07	57.81	62.45	42.13	65.00
Prueba 3	61.22	55.81	62.45	51.21	62.00
Prueba 4	58.21	56.43	62.78	35.23	59.00
Prueba 5	67.61	76.62	69.25	58.21	58.00
Prueba 6	63.45	75.23	65.24	57.15	73.00
Prueba 7	59.21	54.56	74.02	58.12	63.26
Prueba 8	61.42	71.25	65.23	61.07	70.14
Prueba 9	64.23	62.35	71.22	55.41	68.25
Prueba 10	60.23	59.15	63.51	47.26	73.65
Prueba 11	59.45	72.35	60.47	54.48	54.56
Prueba 12	70.11	65.32	65.12	63.54	69.45
Prueba 13	62.45	66.33	63.25	62.25	60.58
Prueba 14	57.12	55.12	71.03	58.17	72.00
Prueba 15	65.23	74.23	66.32	49.23	71.00
Prueba 16	65.32	59.45	65.55	54.32	60.00
Prueba 17	69.18	71.25	64.21	63.25	58.00
Prueba 18	58.52	74.36	67.26	71.02	70.25
Prueba 19	60.25	59.32	66.47	72.45	66.00
Prueba 20	67.12	62.15	68.33	48.35	70.00
Prueba 21	56.45	63.58	67.37	51.25	74.00
Prueba 22	60.35	62.45	68.28	56.25	76.00
Prueba 23	58.36	59.36	67.56	65.12	55.00
Prueba 24	70.11	78.02	69.17	53.36	64.00
Prueba 25	62.25	65.2	69.61	66.54	59.00
Prueba 26	64.41	77.24	70.06	64.25	74.00
Prueba 27	61.36	59.41	73.26	55.26	69.00
Prueba 28	63.45	65.54	65.34	49.32	55.00
Prueba 29	59.48	58.23	67.25	54.32	59.00
Prueba 30	57.26	60.31	68.51	62.52	61.00

**Tabla 30** Utilización de CPU

**Fuente:** Elaboración propia



Luego se obtuvo una sumatoria de todas las tareas por cada prueba:

#### UTILIZACIÓN DE CPU

N° Prueba	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5	Resultados
Prueba 1	59.54	57.26	60.27	44.12	71.00	58.44
Prueba 2	60.07	57.81	62.45	42.13	65.00	57.49
Prueba 3	61.22	55.81	62.45	51.21	62.00	58.54
Prueba 4	58.21	56.43	62.78	35.23	59.00	54.33
Prueba 5	67.61	76.62	69.25	58.21	58.00	65.94
Prueba 6	63.45	75.23	65.24	57.15	73.00	66.81
Prueba 7	59.21	54.56	74.02	58.12	63.26	61.83
Prueba 8	61.42	71.25	65.23	61.07	70.14	65.82
Prueba 9	64.23	62.35	71.22	55.41	68.25	64.29
Prueba 10	60.23	59.15	63.51	47.26	73.65	60.76
Prueba 11	59.45	72.35	60.47	54.48	54.56	60.26
Prueba 12	70.11	65.32	65.12	63.54	69.45	66.71
Prueba 13	62.45	66.33	63.25	62.25	60.58	62.97
Prueba 14	57.12	55.12	71.03	58.17	72.00	62.69
Prueba 15	65.23	74.23	66.32	49.23	71.00	65.20
Prueba 16	65.32	59.45	65.55	54.32	60.00	60.93
Prueba 17	69.18	71.25	64.21	63.25	58.00	65.18
Prueba 18	58.52	74.36	67.26	71.02	70.25	68.28
Prueba 19	60.25	59.32	66.47	72.45	66.00	64.90
Prueba 20	67.12	62.15	68.33	48.35	70.00	63.19
Prueba 21	56.45	63.58	67.37	51.25	74.00	62.53
Prueba 22	60.35	62.45	68.28	56.25	76.00	64.67
Prueba 23	58.36	59.36	67.56	65.12	55.00	61.08
Prueba 24	70.11	78.02	69.17	53.36	64.00	66.93
Prueba 25	62.25	65.20	69.61	66.54	59.00	64.52
Prueba 26	64.41	77.24	70.06	64.25	74.00	69.99
Prueba 27	61.36	59.41	73.26	55.26	69.00	63.66
Prueba 28	63.45	65.54	65.34	49.32	55.00	59.73
Prueba 29	59.48	58.23	67.25	54.32	59.00	59.66
Prueba 30	57.26	60.31	68.51	62.52	61.00	61.92

**Tabla 31** Utilización de CPU – Sumatoria

**Fuente:** Elaboración propia

Por último, se calculó la sumatoria de utilización de CPU de la tarea para todas las pruebas, siendo 1889.25 que viene hacer la variable  $\sum R$  en la matriz de calidad.



#### 5.4.2.5 Métrica: Utilización de la memoria:

Datos obtenidos por cada tarea, haciendo uso de la formula  $X = A$  A= Cantidad de memoria que es usado para realizar una tarea.

UTILIZACIÓN DE LA MEMORIA

	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5
N pruebas	A	A	A	A	A
Prueba 1	64.25	52.45	56.28	51.37	55.62
Prueba 2	55.72	51.66	57.54	52.79	55.16
Prueba 3	56.66	53.78	53.52	53.78	68.46
Prueba 4	59.76	59.66	59.96	52.97	71.09
Prueba 5	58.23	54.92	56.02	74.29	57.13
Prueba 6	57.25	61.25	52.32	57.45	59.12
Prueba 7	56.54	51.26	56.36	62.35	56.36
Prueba 8	62.31	59.32	58.25	60.45	71.26
Prueba 9	65.25	65.36	59.31	58.45	59.48
Prueba 10	59.36	54.26	54.52	55.24	72.28
Prueba 11	59.28	59.36	57.23	56.34	59.38
Prueba 12	58.13	65.15	59.48	57.13	70.56
Prueba 13	57.45	68.12	61.02	52.42	58.34
Prueba 14	54.36	54.35	58.56	51.48	65.25
Prueba 15	62.33	75.36	62.47	55.36	71.28
Prueba 16	59.26	75.26	59.35	78.36	66.33
Prueba 17	63.41	52.36	60.14	57.14	68.48
Prueba 18	57.26	54.16	63.25	54.13	63.54
Prueba 19	60.4	55.47	55.26	55.36	61.15
Prueba 20	59.36	53.26	57.06	59.25	59.35
Prueba 21	56.52	62.35	54.15	59.36	63.47
Prueba 22	57.26	78.29	55.48	60.54	62.04
Prueba 23	59.35	60.56	56.36	59.17	64.58
Prueba 24	61.25	58.25	57.39	58.47	57.26
Prueba 25	65.32	78.36	64.58	51.41	58.45
Prueba 26	55.41	54.46	61.25	56.23	68.11
Prueba 27	63.45	57.63	58.45	74.58	61.2
Prueba 28	62.31	77.56	63.38	59.38	65.58
Prueba 29	61.45	59.29	57.46	63.24	73.41
Prueba 30	59.54	78.26	56.49	65.23	62.14

**Tabla 32** Utilización de la memoria

**Fuente:** Elaboración propia



Luego se obtuvo una sumatoria de todas las tareas por cada prueba:

UTILIZACIÓN DE MEMORIA						
N° Prueba	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5	Resultados
Prueba 1	64.25	52.45	56.28	51.37	55.62	55.99
Prueba 2	55.72	51.66	57.54	52.79	55.16	54.57
Prueba 3	56.66	53.78	53.52	53.78	68.46	57.24
Prueba 4	59.76	59.66	59.96	52.97	71.09	60.69
Prueba 5	58.23	54.92	56.02	74.29	57.13	60.12
Prueba 6	57.25	61.25	52.32	57.45	59.12	57.48
Prueba 7	56.54	51.26	56.36	62.35	56.36	56.57
Prueba 8	62.31	59.32	58.25	60.45	71.26	62.32
Prueba 9	65.25	65.36	59.31	58.45	59.48	61.57
Prueba 10	59.36	54.26	54.52	55.24	72.28	59.13
Prueba 11	59.28	59.36	57.23	56.34	59.38	58.32
Prueba 12	58.13	65.15	59.48	57.13	70.56	62.09
Prueba 13	57.45	68.12	61.02	52.42	58.34	59.47
Prueba 14	54.36	54.35	58.56	51.48	65.25	56.80
Prueba 15	62.33	75.36	62.47	55.36	71.28	65.36
Prueba 16	59.26	75.26	59.35	78.36	66.33	67.71
Prueba 17	63.41	52.36	60.14	57.14	68.48	60.31
Prueba 18	57.26	54.16	63.25	54.13	63.54	58.47
Prueba 19	60.40	55.47	55.26	55.36	61.15	57.53
Prueba 20	59.36	53.26	57.06	59.25	59.35	57.66
Prueba 21	56.52	62.35	54.15	59.36	63.47	59.17
Prueba 22	57.26	78.29	55.48	60.54	62.04	62.72
Prueba 23	59.35	60.56	56.36	59.17	64.58	60.00
Prueba 24	61.25	58.25	57.39	58.47	57.26	58.52
Prueba 25	65.32	78.36	64.58	51.41	58.45	63.62
Prueba 26	55.41	54.46	61.25	56.23	68.11	59.09
Prueba 27	63.45	57.63	58.45	74.58	61.20	63.06
Prueba 28	62.31	77.56	63.38	59.38	65.58	65.64
Prueba 29	61.45	59.29	57.46	63.24	73.41	62.97
Prueba 30	59.54	78.26	56.49	65.23	62.14	64.33

**Tabla 33** Utilización de memoria – sumatoria

**Fuente:** Elaboración propia

Por último, se calculó la sumatoria de utilización de memoria de la tarea para todas las pruebas, siendo 1808.54 que viene hacer la variable  $\sum R$  en la matriz de calidad.

Obtenidos los datos para cada métrica, se aplica a la matriz de calidad:



SUCARACTERITICA	MÉTRICA	FÓRMULA POR PRUEBA	FÓRMULA POR TODAS LAS PRUEBAS	VALOR OBTENIDO (X)	PONDERACION (/10)	VALOR PARCIAL TOTAL (/10)	NIVEL DE IMPORTANCIA	PORCENTAJE DE IMPORTANCIA	VALOR FINAL
comportamiento del tiempo	tiempo de respuesta	$X = B - A$ = Tiempo de envío de petición $B$ = Tiempo en recibir la primera respuesta	$X = \sum R/n$ $n$ = Número de pruebas	$R = 94.89$ $n = 30$ $x = 3.06$	30.63	287.01	A	100%	2.87
	tiempo de espera	$X = B - A$ = Tiempo cuando se inicia un trabajo $B$ = Tiempo en completar el trabajo	$X = \sum R/n$ $n$ = Número de pruebas	$R = 506.37$ $n = 30$ $x = 16.88$	168.79				
	rendimiento	$X = A/T$ $A$ = Número de tareas completadas $T$ = Intervalo de tiempo Donde: $T > 0$	$X = \sum R/n$ $n$ = Número de pruebas	$R = 9.10$ $n = 30$ $x = 0.30$	3.03				
Utilización de recursos	Utilización de CPU	$X = A$ $A$ = cantidad de CPU que es usado para realizar una tarea	$X = \sum R/n$ $n$ = Número de pruebas	$R = 1889.25$ $n = 30$ $x = 62.89$	62.98	287.01	A	100%	2.87
	Utilización de la memoria	$X = A$ $A$ = Cantidad de memoria que es usado para realizar una tarea.	$X = \sum R/n$ $n$ = Número de pruebas	$R = 1808.54$ $n = 30$ $x = 60.28$	60.28				

Tabla 34 Matriz de calidad externa  
Fuente: elaboración propia



### 5.4.1 Aplicación móvil desarrollada con JQuery Mobile:

#### 5.4.1.1 Métrica: Tiempo de respuesta:

Datos obtenidos por cada tarea, haciendo uso de la formula  $X=B - A$  A=Tiempo de envío de petición B= Tiempo en recibir la primera respuesta.

N Pruebas	TIEMPO DE RESPUESTA														
	tarea 1		tarea 2		tarea 3		tarea 4		tarea 5						
	A	B resultado	A	B resultado	A	B resultado	A	B resultado	A	B resultado	A	B resultado			
Prueba 1	42.7	45.5	2.80	46.29	48.23	1.94	28.39	32.62	4.23	38.13	43.32	5.19	48.56	52.45	3.89
Prueba 2	5.58	8.08	2.50	54.12	58.24	4.12	34.06	35.66	1.6	31.31	35.67	4.36	42.26	45.36	3.1
Prueba 3	18.61	21.88	3.27	45.58	48.28	2.7	1.2	3.63	2.43	40.89	42.32	1.43	15.35	18.05	2.7
Prueba 4	12.24	15.45	3.21	36.52	39.26	2.74	20.54	23.41	2.87	15.36	19.59	4.23	21.29	23.41	2.12
Prueba 5	25.36	28.45	3.09	11.26	14.59	3.33	14.21	17.56	3.35	25.16	28.57	3.41	24.21	26.34	2.13
Prueba 6	15.21	18.59	3.38	35.34	38.45	3.11	25.48	28.32	2.84	34.17	37.48	3.31	18.25	20.28	2.03
Prueba 7	9.56	13.24	3.68	42.26	45.28	3.02	35.29	37.51	2.22	26.17	30.24	4.07	36.28	38.29	2.01
Prueba 8	23.54	25.56	2.02	18.25	21.39	3.14	19.38	23.15	3.77	28.45	33.2	4.75	43.07	45.02	1.95
Prueba 9	42.12	45.14	3.02	22.03	25.48	3.45	36.52	39.24	2.72	26.34	30.15	3.81	52.36	54.58	2.22
Prueba 10	51.23	54.03	2.8	19.36	22.18	2.82	42.51	45.32	2.81	10.59	14.26	3.67	36.14	38.49	2.35
Prueba 11	26.35	29.57	3.22	20.15	23.51	3.36	16.23	19.35	3.12	19.35	24.23	4.88	43.16	45.21	2.05
Prueba 12	43.51	46.58	3.07	45.23	48.52	3.29	25.48	28.12	2.64	29.54	34.36	4.82	52.06	55.08	3.02
Prueba 13	14.25	17.34	3.09	24.32	27.41	3.09	35.37	38.54	3.17	35.12	38.56	3.44	29.25	32.18	2.93
Prueba 14	21.25	24.31	3.06	25.36	28.15	2.79	41.35	44.26	2.91	42.17	45.28	3.11	31.26	34.19	2.93
Prueba 15	36.15	38.59	2.44	32.51	35.24	2.73	53.19	56.21	3.02	31.29	34.56	3.27	19.37	22.35	2.98
Prueba 16	28.45	32.24	3.79	51.26	54.19	2.93	16.24	19.23	2.99	35.18	39.09	3.91	11.32	14.16	2.84
Prueba 17	35.42	38.58	3.16	15.28	18.54	3.26	31.26	34.29	3.03	45.21	49.58	4.37	47.15	49.59	2.44
Prueba 18	40.26	43.28	3.02	21.24	24.56	3.32	42.36	45.35	2.99	12.24	15.58	3.34	32.09	36.01	3.92
Prueba 19	22.13	25.38	3.25	19.36	22.55	3.19	39.48	42.59	3.11	13.49	16.57	3.08	33.65	36.11	2.46
Prueba 20	37.15	40.39	3.24	35.24	38.46	3.22	28.42	31.26	2.84	28.16	31.27	3.11	51.36	54.23	2.87
Prueba 21	47.36	50.48	3.12	41.25	45.36	4.11	16.25	19.58	3.33	34.38	38.12	3.74	15.04	18.23	3.19
Prueba 22	19.51	23.09	3.58	31.29	34.28	2.99	8.56	11.24	2.68	46.21	49.59	3.38	37.23	39.52	2.29
Prueba 23	36.54	39.21	2.67	51.38	54.22	2.84	22.15	26.35	4.2	19.38	23.15	3.77	29.07	33.06	3.99
Prueba 24	43.28	46.25	2.97	46.36	49.38	3.02	29.32	33.25	3.93	26.35	29.42	3.07	31.06	34.01	2.95
Prueba 25	52.36	55.19	2.83	16.34	19.21	2.87	46.25	49.4	3.15	16.29	20.04	3.75	35.04	37.12	2.08
Prueba 26	35.28	38.29	3.01	50.28	54.04	3.76	51.23	54.26	3.03	32.47	35.24	2.77	32.11	35.46	3.35
Prueba 27	29.54	32.46	2.92	26.18	29.36	3.18	53.47	56.14	2.67	42.47	46.57	4.1	41.25	43.17	1.92
Prueba 28	16.25	19.37	3.12	47.39	50.24	2.85	19.47	22.48	3.01	34.28	37.59	3.31	38.27	41.14	2.87
Prueba 29	24.18	27.54	3.36	24.25	27.19	2.94	26.59	29.38	2.79	48.37	53.24	4.87	17.24	19.08	1.84
Prueba 30	34.32	37.41	3.09	11.02	14.33	3.31	31.26	34.25	2.99	51.29	55.38	4.09	12.48	14.37	1.89

**Tabla 35** Tiempo de respuesta

Fuente: Elaboración propia



Luego se obtuvo una sumatoria de todas las tareas por cada prueba:

**TIEMPO DE RESPUESTA**

N° Prueba	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5	Resultados
Prueba 1	2.8	1.94	4.23	5.19	3.89	3.61
Prueba 2	2.5	4.12	1.6	4.36	3.1	3.14
Prueba 3	3.27	2.7	2.43	1.43	2.7	2.51
Prueba 4	3.21	2.74	2.87	4.23	2.12	3.03
Prueba 5	3.09	3.33	3.35	3.41	2.13	3.06
Prueba 6	3.38	3.11	2.84	3.31	2.03	2.93
Prueba 7	3.68	3.02	2.22	4.07	2.01	3.00
Prueba 8	2.02	3.14	3.77	4.75	1.95	3.13
Prueba 9	3.02	3.45	2.72	3.81	2.22	3.04
Prueba 10	2.8	2.82	2.81	3.67	2.35	2.89
Prueba 11	3.22	3.36	3.12	4.88	2.05	3.33
Prueba 12	3.07	3.29	2.64	4.82	3.02	3.37
Prueba 13	3.09	3.09	3.17	3.44	2.93	3.14
Prueba 14	3.06	2.79	2.91	3.11	2.93	2.96
Prueba 15	2.44	2.73	3.02	3.27	2.98	2.89
Prueba 16	3.79	2.93	2.99	3.91	2.84	3.29
Prueba 17	3.16	3.26	3.03	4.37	2.44	3.25
Prueba 18	3.02	3.32	2.99	3.34	3.92	3.32
Prueba 19	3.25	3.19	3.11	3.08	2.46	3.02
Prueba 20	3.24	3.22	2.84	3.11	2.87	3.06
Prueba 21	3.12	4.11	3.33	3.74	3.19	3.50
Prueba 22	3.58	2.99	2.68	3.38	2.29	2.98
Prueba 23	2.67	2.84	4.2	3.77	3.99	3.49
Prueba 24	2.97	3.02	3.93	3.07	2.95	3.19
Prueba 25	2.83	2.87	3.15	3.75	2.08	2.94
Prueba 26	3.01	3.76	3.03	2.77	3.35	3.18
Prueba 27	2.92	3.18	2.67	4.1	1.92	2.96
Prueba 28	3.12	2.85	3.01	3.31	2.87	3.03
Prueba 29	3.36	2.94	2.79	4.87	1.84	3.16
Prueba 30	3.09	3.31	2.99	4.09	1.89	3.074

**Tabla 36** Tiempo de respuesta – sumatoria

**Fuente:** elaboración propia

Por último, se calculó la sumatoria de tiempo de respuesta de la tarea para todas las pruebas, siendo 93.47 que viene hacer la variable  $\sum R$  en la matriz de calidad.





### 5.4.1.2 Métrica: Tiempo de espera:

Datos obtenidos por cada tarea, haciendo uso de la fórmula:  $X = B - A$  A=Tiempo cuando se inicia un trabajo  
B=Tiempo en completar el trabajo.

N Pruebas	TIEMPO DE ESPERA														
	tarea 1		tarea 2		tarea 3		tarea 4		tarea 5						
	A	B	resultado	A	B	resultado	A	B	resultado	A	B	resultado			
Prueba 1	1.02	37.97	36.95	45.37	48.43	3.06	51.53	81.25	29.72	38.02	51.28	13.26	25.14	43.26	18.12
Prueba 2	12.91	47.12	34.21	54.1	58.33	4.23	19.86	52.15	32.29	31.31	39.58	8.27	32.18	51.25	19.07
Prueba 3	40.64	60.41	19.77	45.8	49.08	3.28	39.9	75.23	35.33	40.89	53.21	12.32	11.26	22.31	11.05
Prueba 4	9.12	33.05	23.93	36.52	40.26	3.74	20.54	76.28	55.74	15.32	27.26	11.94	13.58	26.25	12.67
Prueba 5	25.36	48.25	22.89	11.26	15.04	3.78	14.21	51.23	37.02	25.16	42.39	17.23	15.27	27.12	11.85
Prueba 6	15.21	32.16	16.95	35.34	38.41	3.07	25.48	62.17	36.69	34.16	54.26	20.1	9.28	23.15	13.87
Prueba 7	9.56	25.37	15.81	42.26	46.58	4.32	35.29	87.41	52.12	26.17	45.29	19.12	25.34	38.21	12.87
Prueba 8	23.54	42.36	18.82	18.25	22.59	4.34	19.38	74.26	54.88	28.44	40.25	11.81	32.06	46.21	14.15
Prueba 9	42.12	62.31	20.19	22.03	25.24	3.21	36.52	93.05	56.53	26.34	42.36	16.02	41.19	55.02	13.83
Prueba 10	51.23	75.36	24.13	19.36	22.51	3.15	42.51	95.26	52.75	28.45	46.37	17.92	24.34	40.31	15.97
Prueba 11	26.35	75.36	49.01	20.15	23.24	3.09	16.23	75.36	59.13	26.34	40.26	13.92	35.12	54.18	19.06
Prueba 12	43.51	62.35	18.84	45.23	49.05	3.82	25.48	84.57	59.09	10.58	30.14	19.56	43.29	59.57	16.28
Prueba 13	14.25	36.58	22.33	24.32	27.51	3.19	35.37	85.03	49.66	19.34	36.25	16.91	21.28	38.17	16.89
Prueba 14	21.25	45.37	24.12	25.36	29.17	3.81	41.35	96.12	54.77	29.54	48.26	18.72	18.36	36.08	17.72
Prueba 15	36.15	61.23	25.08	32.51	35.59	3.08	53.19	94.27	41.08	31.29	51.24	19.95	8.56	28.19	19.63
Prueba 16	28.45	52.36	23.91	51.26	55.14	3.88	16.24	56.12	39.88	35.18	57.36	22.18	7.28	24.25	16.97
Prueba 17	35.42	59.21	23.79	15.28	18.27	2.99	31.26	57.24	25.98	45.21	67.24	22.03	35.41	55.24	19.83
Prueba 18	40.26	61.25	20.99	21.24	24.58	3.34	42.36	98.16	55.8	12.24	32.15	19.91	26.38	46.21	19.83
Prueba 19	22.13	45.21	23.08	19.36	22.58	3.22	39.48	98.24	58.76	13.49	34.23	20.74	26.05	41.06	15.01
Prueba 20	37.15	67.54	30.39	35.24	38.37	3.13	28.42	74.06	45.64	28.16	45.04	16.88	46.27	63.21	16.94
Prueba 21	47.36	79.31	31.95	41.25	44.31	3.06	16.25	75.34	59.09	34.38	49.25	14.87	12.25	30.54	18.29
Prueba 22	19.51	68.25	48.74	31.29	34.52	3.23	8.56	58.35	49.79	46.21	59.27	13.06	27.11	43.27	16.16
Prueba 23	36.54	60.57	24.03	51.38	54.59	3.21	22.15	85.24	63.09	19.38	34.21	14.83	21.06	38.24	17.18
Prueba 24	43.28	71.13	27.85	46.36	49.54	3.18	29.32	87.13	57.81	26.35	40.17	13.82	24.27	41.09	16.82
Prueba 25	52.36	74.59	22.23	16.34	20.28	3.94	46.25	99.53	53.28	16.29	29.37	13.08	26.23	43.01	16.78
Prueba 26	35.28	62.35	27.07	50.28	54.06	3.78	51.23	94.01	42.78	32.47	47.34	14.87	21.25	37.23	15.98
Prueba 27	29.54	57.51	27.97	26.18	29.34	3.16	53.47	94.26	40.79	42.47	56.28	13.81	33.27	51.09	17.82
Prueba 28	16.25	35.21	18.96	47.39	50.37	2.98	19.47	72.15	52.68	34.28	48.57	14.29	24.28	42.05	17.77
Prueba 29	24.18	38.59	14.41	24.25	28.57	4.32	26.59	81.24	54.65	48.37	65.28	16.91	11.24	28.27	17.03
Prueba 30	34.32	58.35	24.03	11.02	14.05	3.03	31.26	91.03	59.77	51.29	67.24	15.95	7.28	24.33	17.05

**Tabla 37** Tiempo de espera  
**Fuente:** Elaboración propia



Luego se obtuvo una sumatoria de todas las tareas por cada prueba:

TIEMPO DE ESPERA						
N° Prueba	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5	Resultados
Prueba 1	36.95	29.72	29.72	13.26	18.12	25.55
Prueba 2	34.21	32.29	32.29	8.27	19.07	25.23
Prueba 3	19.77	35.33	35.33	12.32	11.05	22.76
Prueba 4	23.93	55.74	55.74	11.94	12.67	32.00
Prueba 5	22.89	37.02	37.02	17.23	11.85	25.20
Prueba 6	16.95	36.69	36.69	20.1	13.87	24.86
Prueba 7	15.81	52.12	52.12	19.12	12.87	30.41
Prueba 8	18.82	54.88	54.88	11.81	14.15	30.91
Prueba 9	20.19	56.53	56.53	16.02	13.83	32.62
Prueba 10	24.13	52.75	52.75	17.92	15.97	32.70
Prueba 11	49.01	59.13	59.13	13.92	19.06	40.05
Prueba 12	18.84	59.09	59.09	19.56	16.28	34.57
Prueba 13	22.33	49.66	49.66	16.91	16.89	31.09
Prueba 14	24.12	54.77	54.77	18.72	17.72	34.02
Prueba 15	25.08	41.08	41.08	19.95	19.63	29.36
Prueba 16	23.91	39.88	39.88	22.18	16.97	28.56
Prueba 17	23.79	25.98	25.98	22.03	19.83	23.52
Prueba 18	20.99	55.8	55.8	19.91	19.83	34.47
Prueba 19	23.08	58.76	58.76	20.74	15.01	35.27
Prueba 20	30.39	45.64	45.64	16.88	16.94	31.10
Prueba 21	31.95	59.09	59.09	14.87	18.29	36.66
Prueba 22	48.74	49.79	49.79	13.06	16.16	35.51
Prueba 23	24.03	63.09	63.09	14.83	17.18	36.44
Prueba 24	27.85	57.81	57.81	13.82	16.82	34.82
Prueba 25	22.23	53.28	53.28	13.08	16.78	31.73
Prueba 26	27.07	42.78	42.78	14.87	15.98	28.70
Prueba 27	27.97	40.79	40.79	13.81	17.82	28.24
Prueba 28	18.96	52.68	52.68	14.29	17.77	31.28
Prueba 29	14.41	54.65	54.65	16.91	17.03	31.53
Prueba 30	24.03	59.77	59.77	15.95	17.05	35.31

**Tabla 38** Tiempo de espera – sumatoria  
**Fuente:** Elaboración propia

Por último, se calculó la sumatoria de tiempo de respuesta de la tarea para todas las pruebas, siendo 934.48 que viene hacer la variable  $\sum R$  en la matriz de calidad.



### 5.4.1.3 Métrica: Rendimiento:

Datos obtenidos por cada tarea, haciendo uso de la fórmula:  $X=A/T$  A=Número de tareas completadas  
T=Intervalo de tiempo Donde:  $T>0$

	TAREA 1	TAREA 2	TAREA 3	TAREA 4	TAREA 5
N° Pruebas	T	T	T	T	T
Prueba 1	36.95	3.06	29.72	13.26	18.12
Prueba 2	34.21	4.23	32.29	8.27	19.07
Prueba 3	19.77	3.28	35.33	12.32	11.05
Prueba 4	23.93	3.74	55.74	11.94	12.67
Prueba 5	22.89	3.78	37.02	17.23	11.85
Prueba 6	16.95	3.07	36.69	20.1	13.87
Prueba 7	15.81	4.32	52.12	19.12	12.87
Prueba 8	18.82	4.34	54.88	11.81	14.15
Prueba 9	20.19	3.21	56.53	16.02	13.83
Prueba 10	24.13	3.15	52.75	17.92	15.97
Prueba 11	49.01	3.09	59.13	13.92	19.06
Prueba 12	18.84	3.82	59.09	19.56	16.28
Prueba 13	22.33	3.19	49.66	16.91	16.89
Prueba 14	24.12	3.81	54.77	18.72	17.72
Prueba 15	25.08	3.08	41.08	19.95	19.63
Prueba 16	23.91	3.88	39.88	22.18	16.97
Prueba 17	23.79	2.99	25.98	22.03	19.83
Prueba 18	20.99	3.34	55.8	19.91	19.83
Prueba 19	23.08	3.22	58.76	20.74	15.01
Prueba 20	30.39	3.13	45.64	16.88	16.94
Prueba 21	31.95	3.06	59.09	14.87	18.29
Prueba 22	48.74	3.23	49.79	13.06	16.16
Prueba 23	24.03	3.21	63.09	14.83	17.18
Prueba 24	27.85	3.18	57.81	13.82	16.82
Prueba 25	22.23	3.94	53.28	13.08	16.78
Prueba 26	27.07	3.78	42.78	14.87	15.98
Prueba 27	27.97	3.16	40.79	13.81	17.82
Prueba 28	18.96	2.98	52.68	14.29	17.77
Prueba 29	14.41	4.32	54.65	16.91	17.03
Prueba 30	24.03	3.03	59.77	15.95	17.05

**Tabla 39** Tiempo de Rendimiento

**Fuente:** Elaboración propia



Luego se obtuvo una sumatoria de todas las tareas por cada prueba:

TIEMPO DE RENDIMIENTO						
N° Prueba	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5	Resultados
Prueba 1	36.95	3.06	29.72	13.26	18.12	0.25
Prueba 2	34.21	4.23	32.29	8.27	19.07	0.25
Prueba 3	19.77	3.28	35.33	12.32	11.05	0.31
Prueba 4	23.93	3.74	55.74	11.94	12.67	0.23
Prueba 5	22.89	3.78	37.02	17.23	11.85	0.27
Prueba 6	16.95	3.07	36.69	20.1	13.87	0.28
Prueba 7	15.81	4.32	52.12	19.12	12.87	0.24
Prueba 8	18.82	4.34	54.88	11.81	14.15	0.24
Prueba 9	20.19	3.21	56.53	16.02	13.83	0.23
Prueba 10	24.13	3.15	52.75	17.92	15.97	0.22
Prueba 11	49.01	3.09	59.13	13.92	19.06	0.17
Prueba 12	18.84	3.82	59.09	19.56	16.28	0.21
Prueba 13	22.33	3.19	49.66	16.91	16.89	0.23
Prueba 14	24.12	3.81	54.77	18.72	17.72	0.21
Prueba 15	25.08	3.08	41.08	19.95	19.63	0.23
Prueba 16	23.91	3.88	39.88	22.18	16.97	0.23
Prueba 17	23.79	2.99	25.98	22.03	19.83	0.26
Prueba 18	20.99	3.34	55.8	19.91	19.83	0.21
Prueba 19	23.08	3.22	58.76	20.74	15.01	0.21
Prueba 20	30.39	3.13	45.64	16.88	16.94	0.22
Prueba 21	31.95	3.06	59.09	14.87	18.29	0.20
Prueba 22	48.74	3.23	49.79	13.06	16.16	0.19
Prueba 23	24.03	3.21	63.09	14.83	17.18	0.20
Prueba 24	27.85	3.18	57.81	13.82	16.82	0.21
Prueba 25	22.23	3.94	53.28	13.08	16.78	0.23
Prueba 26	27.07	3.78	42.78	14.87	15.98	0.24
Prueba 27	27.97	3.16	40.79	13.81	17.82	0.24
Prueba 28	18.96	2.98	52.68	14.29	17.77	0.23
Prueba 29	14.41	4.32	54.65	16.91	17.03	0.23
Prueba 30	24.03	3.03	59.77	15.95	17.05	0.21

**Tabla 40** Tiempo de rendimiento – sumatoria

**Fuente:** Elaboración propia

Por último, se calculó la sumatoria de rendimiento de la tarea para todas las pruebas, siendo 6.89 que viene hacer la variable  $\sum R$  en la matriz de calidad.



5.4.1.4 Métrica: Utilización de CPU:

Datos obtenidos por cada tarea, haciendo uso de la formula  $X = A$  A= cantidad de CPU que es usado para realizar una tarea.

UTILIZACIÓN DE CPU

	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5
N Pruebas	A	A	A	A	A
Prueba 1	73.32	47.75	66.65	59.96	49.25
Prueba 2	73.03	55.81	79.65	37.09	51.26
Prueba 3	67.1	71.88	58.57	45.21	62.14
Prueba 4	65.14	72.32	64.05	50.23	47.28
Prueba 5	71.23	67.43	63.25	41.45	52.19
Prueba 6	75.14	65.35	69.39	55.28	54.06
Prueba 7	64.25	70.26	76.21	53.21	51.02
Prueba 8	73.52	68.35	71.56	42.13	48.26
Prueba 9	60.35	62.14	68.16	43.21	74.06
Prueba 10	75.36	59.45	74.45	50.11	61.02
Prueba 11	68.23	62.13	78.28	52.47	51.2
Prueba 12	75.54	69.27	79.04	56.25	47.31
Prueba 13	71.27	70.38	66.18	47.56	49.34
Prueba 14	69.54	58.45	67.16	54.24	53.04
Prueba 15	67.58	73.25	73.25	49.36	51.06
Prueba 16	65.23	65.25	59.24	48.36	46.28
Prueba 17	71.14	71.28	72.39	53.28	48.34
Prueba 18	72.05	68.12	65.28	55.47	57.12
Prueba 19	68.16	65.19	69.17	51.36	60.07
Prueba 20	73.06	73.14	58.54	45.27	47.26
Prueba 21	69.02	75.19	67.18	48.26	55.08
Prueba 22	67.18	64.09	65.29	53.18	46.17
Prueba 23	68.27	59.07	72.26	46.19	47.16
Prueba 24	72.41	62.34	79.26	52.08	52.13
Prueba 25	73.16	63.41	67.23	47.15	61.08
Prueba 26	75.21	73.21	70.38	47.25	52.11
Prueba 27	67.15	76.18	64.21	45.17	47.05
Prueba 28	76.05	67.21	67.48	59.41	48.35
Prueba 29	68.19	63.16	68.46	53.29	53.27
Prueba 30	70.26	57.18	72.05	54.23	51.06

**Tabla 41** Utilización de CPU  
**Fuente:** Elaboración propia



Luego se obtuvo una sumatoria de todas las tareas por cada prueba:

**UTILIZACIÓN DE CPU**

	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5	Resultados
Prueba 1	73.32	45.75	66.65	66.65	49.25	60.32
Prueba 2	73.03	53.81	79.65	79.65	51.26	67.48
Prueba 3	67.1	69.88	58.57	58.57	62.14	63.25
Prueba 4	65.14	70.32	64.05	64.05	47.28	62.17
Prueba 5	71.23	65.43	63.25	63.25	52.19	63.07
Prueba 6	75.14	63.35	69.39	69.39	54.06	66.27
Prueba 7	64.25	68.26	76.21	76.21	51.02	67.19
Prueba 8	73.52	66.35	71.56	71.56	48.26	66.25
Prueba 9	60.35	60.14	68.16	68.16	74.06	66.17
Prueba 10	75.36	57.45	74.45	74.45	61.02	68.55
Prueba 11	68.23	60.13	78.28	78.28	51.2	67.22
Prueba 12	75.54	67.27	79.04	79.04	47.31	69.64
Prueba 13	71.27	68.38	66.18	66.18	49.34	64.27
Prueba 14	69.54	56.45	67.16	67.16	53.04	62.67
Prueba 15	67.58	71.25	73.25	73.25	51.06	67.28
Prueba 16	65.23	63.25	59.24	59.24	46.28	58.65
Prueba 17	71.14	69.28	72.39	72.39	48.34	66.71
Prueba 18	72.05	66.12	65.28	65.28	57.12	65.17
Prueba 19	68.16	63.19	69.17	69.17	60.07	65.95
Prueba 20	73.06	71.14	58.54	58.54	47.26	61.71
Prueba 21	69.02	73.19	67.18	67.18	55.08	66.33
Prueba 22	67.18	62.09	65.29	65.29	46.17	61.20
Prueba 23	68.27	57.07	72.26	72.26	47.16	63.40
Prueba 24	72.41	60.34	79.26	79.26	52.13	68.68
Prueba 25	73.16	61.41	67.23	67.23	61.08	66.02
Prueba 26	75.21	71.21	70.38	70.38	52.11	67.86
Prueba 27	67.15	74.18	64.21	64.21	47.05	63.36
Prueba 28	76.05	65.21	67.48	67.48	48.35	64.91
Prueba 29	68.19	61.16	68.46	68.46	53.27	63.91
Prueba 30	70.26	55.18	72.05	72.05	51.06	64.12

**Tabla 42** Utilización de CPU – sumatoria  
**Fuente:** Elaboración propia

Por último, se calculó la sumatoria de utilización de CPU de la tarea para todas las pruebas, siendo 1949.79 que viene hacer la variable  $\sum R$  en la matriz de calidad.



#### 5.4.1.5 Métrica: Utilización de la memoria:

Datos obtenidos por cada tarea, haciendo uso de la formula  $X = A$  A= Cantidad de memoria que es usado para realizar una tarea.

UTILIZACIÓN DE LA MEMORIA

	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5
N° Pruebas	A	A	A	A	A
Prueba 1	45.64	49.18	72.22	77.21	65.24
Prueba 2	70.79	57.5	75.87	66.38	68.11
Prueba 3	71.72	74.77	75.25	75.33	65.18
Prueba 4	68.45	62.23	72.35	71.16	71.24
Prueba 5	65.35	59.25	76.21	76.27	78.35
Prueba 6	55.58	61.21	74.19	68.14	68.23
Prueba 7	78.54	57.24	75.38	65.28	61.25
Prueba 8	68.24	51.45	69.28	68.49	66.12
Prueba 9	70.78	60.23	68.49	71.19	76.21
Prueba 10	75.24	64.12	76.15	78.26	62.39
Prueba 11	77.56	63.24	75.21	69.19	73.19
Prueba 12	56.14	61.21	78.13	73.29	68.16
Prueba 13	68.35	58.24	74.36	77.18	78.21
Prueba 14	59.41	53.29	68.25	78.19	65.28
Prueba 15	68.35	66.34	63.24	76.39	72.03
Prueba 16	58.47	58.37	74.25	69.21	64.28
Prueba 17	62.31	64.58	76.58	77.23	63.27
Prueba 18	68.54	71.26	72.48	78.25	72.19
Prueba 19	71.25	67.18	69.25	74.28	62.38
Prueba 20	76.21	65.25	71.24	71.38	71.04
Prueba 21	69.41	68.24	78.37	76.29	67.03
Prueba 22	73.54	64.38	77.19	71.18	66.25
Prueba 23	68.25	67.26	68.18	66.17	73.26
Prueba 24	73.45	59.23	76.31	70.29	71.08
Prueba 25	67.37	64.57	79.12	71.49	65.29
Prueba 26	69.38	68.35	68.57	64.25	65.31
Prueba 27	74.25	65.38	65.58	72.38	65.28
Prueba 28	59.58	69.48	76.58	68.14	65.23
Prueba 29	65.57	71.24	68.27	63.59	58.25
Prueba 30	71.24	68.21	75.26	69.28	59.14

**Tabla 43** Utilización de memoria

**Fuente:** Elaboración propia



Luego se obtuvo una sumatoria de todas las tareas por cada prueba:

UTILIZACIÓN DE MEMORIA						
N° Prueba	tarea 1	tarea 2	tarea 3	tarea 4	tarea 5	Resultados
Prueba 1	45.64	49.18	72.22	77.21	65.24	61.90
Prueba 2	70.79	57.50	75.87	66.38	68.11	67.73
Prueba 3	71.72	74.77	75.25	75.33	65.18	72.45
Prueba 4	68.45	62.23	72.35	71.16	71.24	69.09
Prueba 5	65.35	59.25	76.21	76.27	78.35	71.09
Prueba 6	55.58	61.21	74.19	68.14	68.23	65.47
Prueba 7	78.54	57.24	75.38	65.28	61.25	67.54
Prueba 8	68.24	51.45	69.28	68.49	66.12	64.72
Prueba 9	70.78	60.23	68.49	71.19	76.21	69.38
Prueba 10	75.24	64.12	76.15	78.26	62.39	71.23
Prueba 11	77.56	63.24	75.21	69.19	73.19	71.68
Prueba 12	56.14	61.21	78.13	73.29	68.16	67.39
Prueba 13	68.35	58.24	74.36	77.18	78.21	71.27
Prueba 14	59.41	53.29	68.25	78.19	65.28	64.88
Prueba 15	68.35	66.34	63.24	76.39	72.03	69.27
Prueba 16	58.47	58.37	74.25	69.21	64.28	64.92
Prueba 17	62.31	64.58	76.58	77.23	63.27	68.79
Prueba 18	68.54	71.26	72.48	78.25	72.19	72.54
Prueba 19	71.25	67.18	69.25	74.28	62.38	68.87
Prueba 20	76.21	65.25	71.24	71.38	71.04	71.02
Prueba 21	69.41	68.24	78.37	76.29	67.03	71.87
Prueba 22	73.54	64.38	77.19	71.18	66.25	70.51
Prueba 23	68.25	67.26	68.18	66.17	73.26	68.62
Prueba 24	73.45	59.23	76.31	70.29	71.08	70.07
Prueba 25	67.37	64.57	79.12	71.49	65.29	69.57
Prueba 26	69.38	68.35	68.57	64.25	65.31	67.17
Prueba 27	74.25	65.38	65.58	72.38	65.28	68.57
Prueba 28	59.58	69.48	76.58	68.14	65.23	67.80
Prueba 29	65.57	71.24	68.27	63.59	58.25	65.38
Prueba 30	71.24	68.21	75.26	69.28	59.14	68.63

**Tabla 44** Utilización de memoria – sumatoria

**Fuente:** Elaboración propia

Por último, se calculó la sumatoria de utilización de memoria de la tarea para todas las pruebas, siendo 2059.42 que viene hacer la variable  $\sum R$  en la matriz de calidad.

Obtenidos los datos para cada métrica, se aplica a la matriz de calidad:





SUB CARACTERÍSTICA	MÉTRICA	FÓRMULA POR PRUEBA	formula por todas las pruebas	VALOR OBTENIDO (X)	PONDERACION (/10)	VALOR PARCIAL TOTAL (/10)	NIVEL DE IMPORTANCIA	% DE IMPORTANCIA	VALOR FINAL
comportamiento del tiempo	tiempo de respuesta	$X = B - A = \text{Tiempo de envío de petición}$ $B = \text{Tiempo en recibir la primera respuesta}$	$X = \sum R/n$ $n = \text{Número de pruebas}$ $X = 3.12$	R=93.47 n=30 X=3.12	31.16	336.27	A	100%	3.36
	tiempo de espera	$X = B - A = \text{Tiempo cuando se inicia un trabajo}$ $B = \text{Tiempo en completar el trabajo}$	$X = \sum R/n$ $n = \text{Número de pruebas}$ $X = 31.15$	R=934.48 n=30 X=31.15	311.49				
	rendimiento	$X = A/T$ A=Número de tareas completadas T=Intervalo de tiempo Donde: $T > 0$	$X = \sum R/n$ $n = \text{Número de pruebas}$ $X = 0.23$	R=6.89 n=30 X=0.23	2.30				
Utilización de recursos	Utilización de CPU	$X = A$ A= cantidad de CPU que es usado para realizar una tarea	$X = \sum R/n$ $n = \text{Número de pruebas}$ $x = 64.69$	R=1949.79 n=30 x=64.69	649.99	336.27	A	100%	3.36
	Utilización de la memoria	$X = A = \text{Cantidad de memoria que es usado para realizar una tarea.}$	$X = \sum R/n$ $n = \text{Número de pruebas}$ $x = 68.65$	R=2059 n=30 x=68.65	686.65				

Tabla 45 Matriz de calidad externa  
Fuente: Elaboración propia



## CONCLUSIONES Y RECOMENDACIONES

### 6.1. Conclusiones

- La investigación realizada por la compañía (G2 Crowd, 2017) ayudó en gran medida a determinar cuáles son los frameworks de desarrollo móvil líderes en el año 2017, reduciendo significativamente el tiempo y esfuerzo necesario para la selección de los frameworks para la evaluación.
- De acuerdo a investigaciones sobre la norma ISO 25000, se estableció los parámetros de evaluación para medir el rendimiento, seleccionando sus respectivas fórmulas para métrica.
- El desarrollo de la aplicación “ADP EAT” permitió evaluar de forma real y directa el rendimiento de los frameworks, bajo la norma ISO 25000.
- En la evaluación de comportamiento en el tiempo se analizaron las métricas de tiempo de respuesta, y tiempo de espera y en utilización de recursos las métricas de rendimiento, uso de CPU y memoria RAM, pertenecientes a la característica eficiencia en el desempeño. Los resultados de esta evaluación sugieren que: el comportamiento en el tiempo de la aplicación móvil desarrollada con Apache cordova obtiene el menor valor, y la utilización de recursos tanto de memoria como de CPU y rendimiento, también es menor en la aplicación desarrollada con framework Apache cordova que obtuvo 2.87 frente al framework jQuery Mobile con 3.36. Tras la confrontación de los dos frameworks de desarrollo de aplicaciones móviles mediante los atributos de la calidad relacionados con la eficiencia en el desempeño, se confirmó que el desarrollo con Framework Apache cordova genera aplicaciones de mejor rendimiento que el desarrollo basado en framework jQuery Mobile.



## 6.2. Recomendaciones

- Hacer una correcta selección de la herramienta al momento de desarrollar una aplicación móvil, para poder evitar inconvenientes que requieran de un mayor coste de tiempo y esfuerzo.
- Investigar constantemente las mejoras que presentan los frameworks por medio de la publicación de sus actualizaciones, así como también de la aparición de nuevos frameworks; con el fin de aprovechar al máximo las prestaciones que este tipo de herramientas ofrecen.
- En lo que respecta al análisis de rendimiento y monitorización sería conveniente tener acceso a otros dispositivos Android con diferentes especificaciones de hardware en cuanto a pantalla, memoria y CPU, y distintas versiones del sistema operativo.
- La comparación de los dos frameworks de desarrollo, se llevó a cabo desde la perspectiva de los atributos de calidad relacionados con la eficiencia, sin embargo, resulta importante para trabajos futuros considerar una evaluación a las aplicaciones donde se incluya usabilidad, funcionalidad y seguridad.



## REFERENCIAS

- Aguado, J.-M., Martínez, I. J., & Cañete-Sanz, L. (2015). Tendencias evolutivas del contenido digital en aplicaciones móviles. *El Profesional de La Información*, 24(6), 787. <https://doi.org/10.3145/epi.2015.nov.10>
- Amparo, B. C. E. (2014). *Evaluación de calidad de productos software en empresas de desarrollo de software aplicando la norma ISO/IEC 25000*.
- Andrés, D. D. J. (2014). *ESTUDIO COMPARATIVO DE DIFERENTES FRAMEWORKS DE ALTO NIVEL PARA EL DESARROLLO DE APLICACIONES MOVILES EN ANDROID*.
- Bender, A. G., & Chavez, A. D. (2014). Desarrollo de aplicaciones híbridas multiplataforma con PhoneGap para dispositivos HandHeld.
- BOLIVAR MUÑOZ, C. E. (2016). ELABORACIÓN DE UNA GUÍA DE APLICACIÓN DE MÉTRICAS DE CALIDAD PARA METODOLOGÍAS ÁGILES DE DESARROLLO WEB EN WWW.MACHALAMOVIL.COM.
- Butler, M. (2011). Android: Changing the mobile landscape. *IEEE Pervasive Computing*, 10(1), 4–7. <https://doi.org/10.1109/MPRV.2011.1>
- Cuello, J., & Vittone, J. (2014). *Diseñando apps para móviles* (Catalina D). Retrieved from [http://cbueg-mt.iii.com/iii/encore/record/C\\_\\_Rb6237507\\_\\_Sdisseny aplicació mòbil\\_\\_Ff:facetcloud:disseny:disseny:disseny::\\_\\_Orightresult\\_\\_U\\_\\_X1?lang=cat&suite=def](http://cbueg-mt.iii.com/iii/encore/record/C__Rb6237507__Sdisseny aplicació mòbil__Ff:facetcloud:disseny:disseny:disseny::__Orightresult__U__X1?lang=cat&suite=def)
- Guncay Barzallo, M. C., & Samaniego Mosquera, J. W. (2015). *Evaluación de frameworks para el desarrollo de aplicaciones móviles mltiplataforma*.
- Hidalgo Orozco, V. A., & Casigña Parra, R. V. (2015). *Análisis del rendimiento entre ios y android para desarrollar el sistema de posicionamiento y administración para taxis ejecutivos. Statewide Agricultural Land Use Baseline 2015*. <https://doi.org/10.1017/CBO9781107415324.004>
- ISO 25010. (2017). Retrieved December 11, 2017, from <http://iso25000.com/index.php/normas-iso-25000/iso-25010>
- Kundu, T. K., & Paul, K. (2016). Improving Android performance and energy efficiency. *Proceedings of the IEEE International Conference on VLSI Design*, 256–261. <https://doi.org/10.1109/VLSID.2011.63>
- La, H. J., & Kim, S. D. (2012). Balanced MVC architecture for high efficiency mobile applications. *KSII Transactions on Internet and Information Systems*, 6(5), 1421–1444. <https://doi.org/10.3837/tiis.2012.05.0010>
- Liu, Y., Xu, C., & Cheung, S. C. (2015). Diagnosing energy efficiency and performance for mobile internetware applications. *IEEE Software*, 32(1), 67–75. <https://doi.org/10.1109/MS.2015.4>
- Luna, F. (2014). *Desarrollo web para dispositivos móviles: Herramientas para diseñar y programar WebApps* (Fox Andina S.A.). Retrieved from [https://books.google.com.pe/books?id=BRSRDAAAQBAJ&printsec=frontcover&dq=Desarrollo+Web+para+Dispositivos+Móviles+Luna,+Fernando&hl=es-419&sa=X&ved=0ahUKEwivmc\\_Z1KbXAhWI7iYKHRvLBFcQ6AEIJDA#v=onepage&q&f=false](https://books.google.com.pe/books?id=BRSRDAAAQBAJ&printsec=frontcover&dq=Desarrollo+Web+para+Dispositivos+Móviles+Luna,+Fernando&hl=es-419&sa=X&ved=0ahUKEwivmc_Z1KbXAhWI7iYKHRvLBFcQ6AEIJDA#v=onepage&q&f=false)



se

Pérez Reyes, A. (2014). *Análisis de rendimiento de aplicaciones Android*. Universidad de la Laguna.

Prateek Aggarwal. (n.d.). Agile Scrum Methodology for Mobile Apps - Innovapptive Inc. Retrieved July 8, 2017, from <http://www.innovapptive.com/blog/agile-scrum-methodology-for-mobile-apps/>

Raudjärv, R., & Gregersen, A. R. (2015). JRebel . Android : Runtime Class- and Resource Reloading for Android, 741–745. <https://doi.org/10.1109/ICSE.2015.337>

Ravi Tamada. (n.d.). Inicio de sesión y registro de Android con PHP, MySQL y SQLite - AndroidHive. Retrieved from <http://www.androidhive.info/2012/01/android-login-and-registration-with-php-mysql-and-sqlite/>

Rengifo, F., & Betancourt, C. (2011). *Frameworks Y Herramientas Para El Desarrollo De Aplicaciones Orientadas a Dispositivos Moviles*. UNIVERSIDAD TECNOLÓGICA DE PEREIRA.

Sanjuán, Á. J. D., Rodríguez, J. L. P., & Rosado, A. A. (2015). Comparación de dos tecnologías de desarrollo de aplicaciones móviles desde la perspectiva del rendimiento como atributo de calidad. *Scientia Et Technica*, 20(1), 81–87. <https://doi.org/10.18046/syt.v14i39.2347>

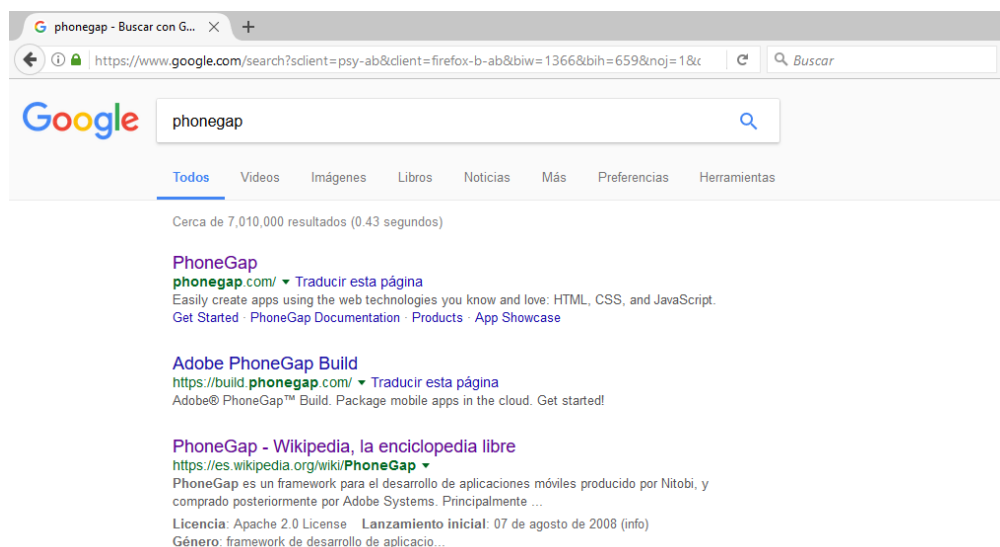
Syer, M. D., Nagappan, M., Adams, B., & Hassan, A. E. (2015). Studying the relationship between source code quality and mobile platform dependence. *Software Quality Journal*, 23(3), 485–508. <https://doi.org/10.1007/s11219-014-9238-2>



## ANEXO 1

### Phonegap

- **Desarrollo multiplataforma:** permite crear aplicaciones móviles multiplataforma a partir de tecnologías web como HTML5, CSS3 y Javascript. Hasta la versión 2.1.0 de PhoneGap, las plataformas soportadas son las siguientes: Android, Blackberry, iOS, Symbian, WebOS, Windows Phone 7 y 8, Windows 8, Bada, Tizen.
- **Edad de software:** 07 de agosto de 2008.
- **Costo de software:** es un framework gratuito.
- **Open source:** si
- **Documentación:** <http://phonegap.com/>
- **Popularidad:**

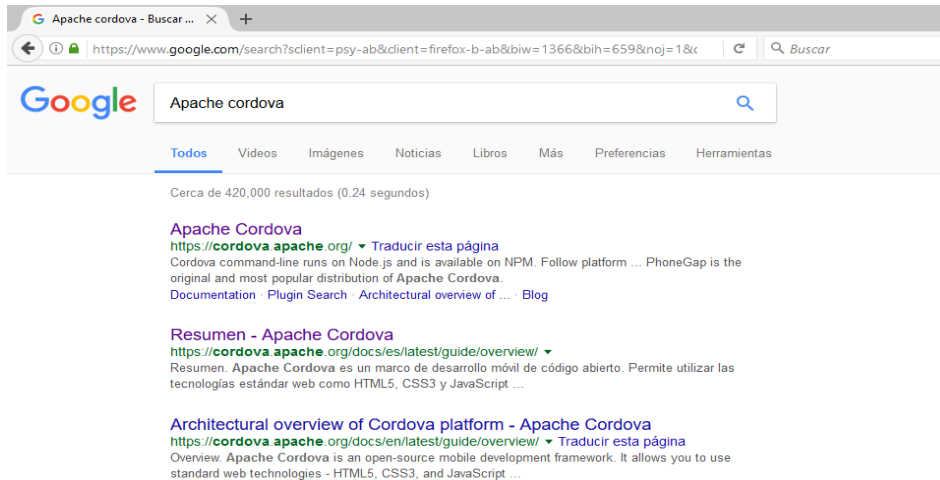


### Apache cordova

- **Desarrollo multiplataforma:** Permite utilizar las tecnologías estándar web como HTML5, CSS3 y JavaScript para desarrollo multiplataforma. Android, blackberry10, IOS, Ubuntu, WP8 (Windows Phone 8), Windows (8.1, 10, Teléfono 8.1), OS X. <https://cordova.apache.org/>
- **Edad de software:** 2011.
- **Costo de software:** gratuito
- **Open source:** Cordova es un marco de desarrollo móvil de código abierto.



- **Documentación:** <https://cordova.apache.org/>
- **Popularidad:**



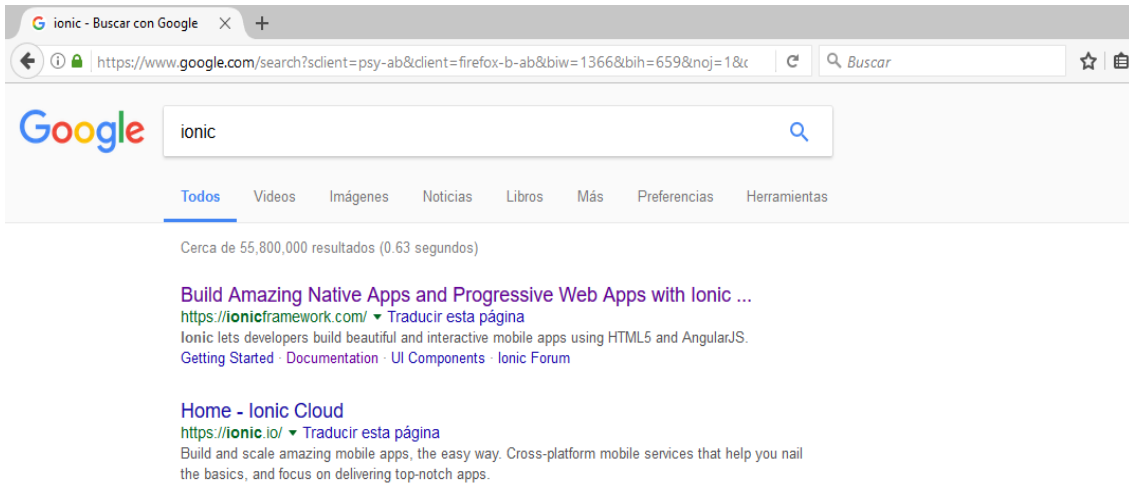
## Ionic

- **Desarrollo multiplataforma:** diseñado para crear aplicaciones móviles en múltiples plataformas. Android, iOS, Amazon Fire OS, Blackberry 10, navegador, Firefox OS, Ubuntu, WebOS, Windows Phone 8 y Windows.
- **Edad de software:** Ionic fue creado por Max Lynch, Ben Sperry, y Adam Bradley de Drifty Co. en 2013.

<b>Versión inicial</b>	2013
<b>versión estable</b>	2.0.0/ 14 de diciembre de, el año 2016; hace 2 meses

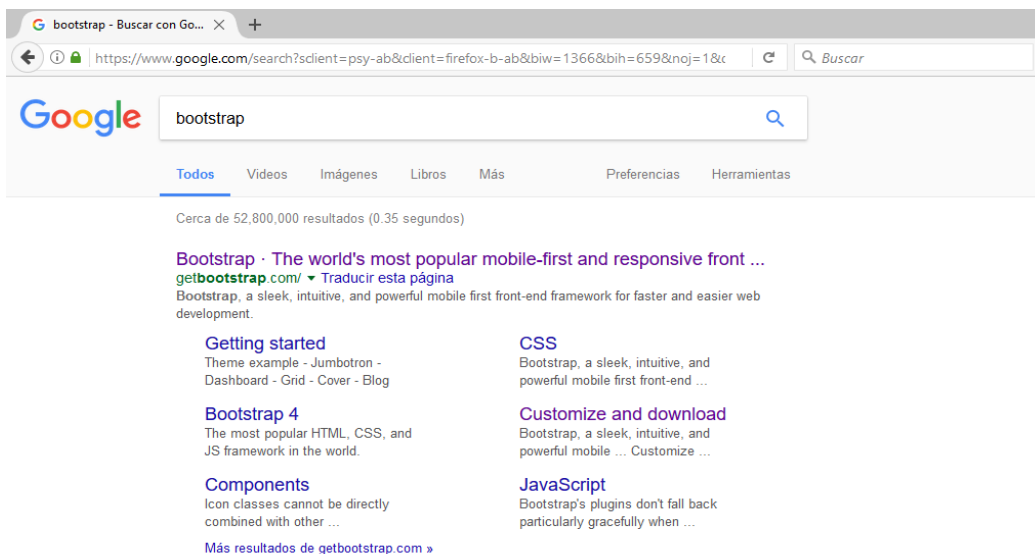
- **Costo del software:** Ionic es una herramienta gratuita.
- **Open source:** Ionic es una herramienta open source, para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JS. Está construido con Sass y optimizado con AngularJS.
- **Documentación:** <https://ionic.io/>
- **Popularidad:**





## Bootstrap

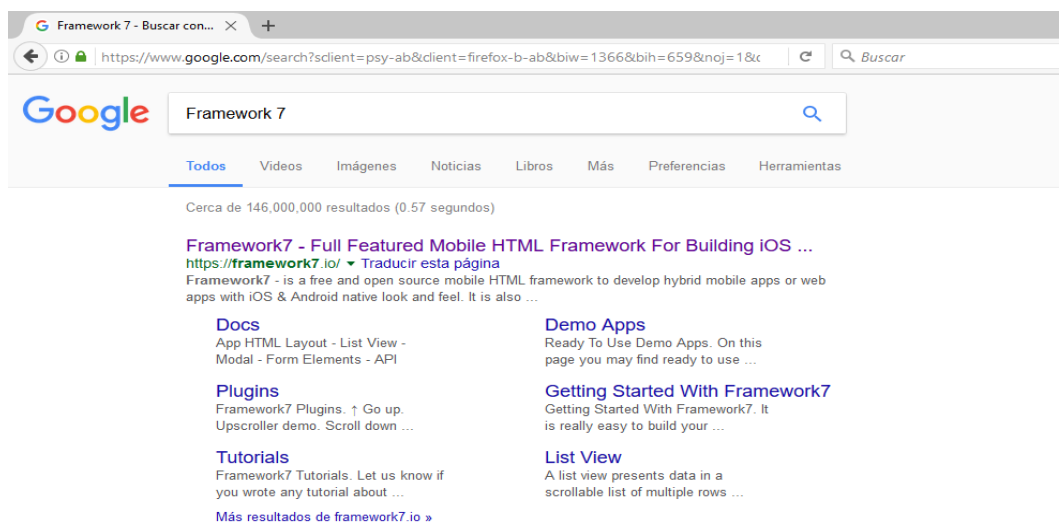
- **Desarrollo multiplataforma:** Android, iOS, MacOS x , Windows (http://librosweb.es/libro/bootstrap\_3/capitulo\_1/compatibilidad\_con\_los\_navegadores.html)
- **Edad de software:** Agosto 2011.
- **Costo de software:** gratuito
- **Open source:** es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web.
- **Documentación:** http://getbootstrap.com/
- **Popularidad:**





## Framework 7

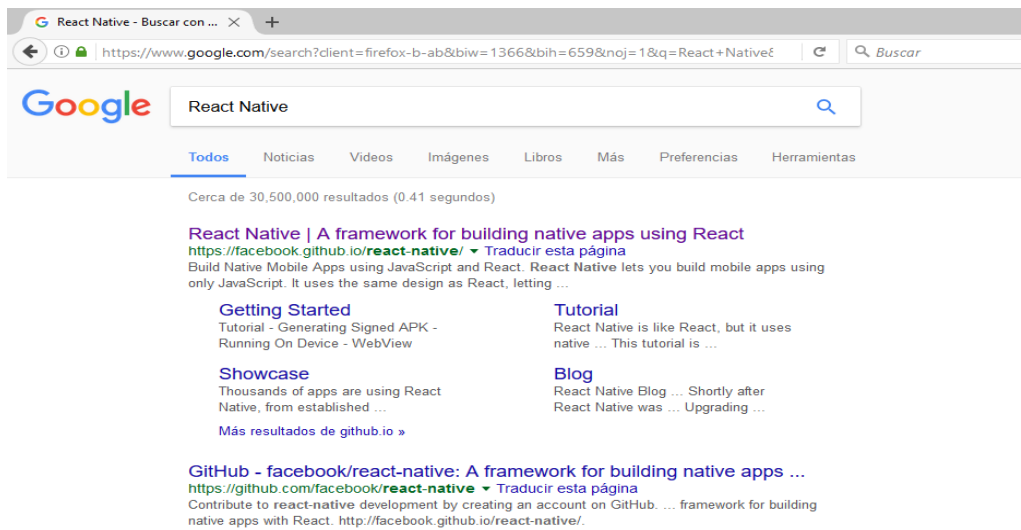
- **Desarrollo multiplataforma:** iOS y Android.  
(<http://www.phonegapSpain.com/introduccion-a-framework7/>)
- **Edad de software:** Framework7 se introdujo en el año 2014. La última versión 1.4.2 fue liberado de febrero el año 2016 bajo licencia MIT.
- **Costo de software:** gratuito
- **Open source:** si
- **Documentación:** <https://framework7.io/>
- **Popularidad:**



## React Native

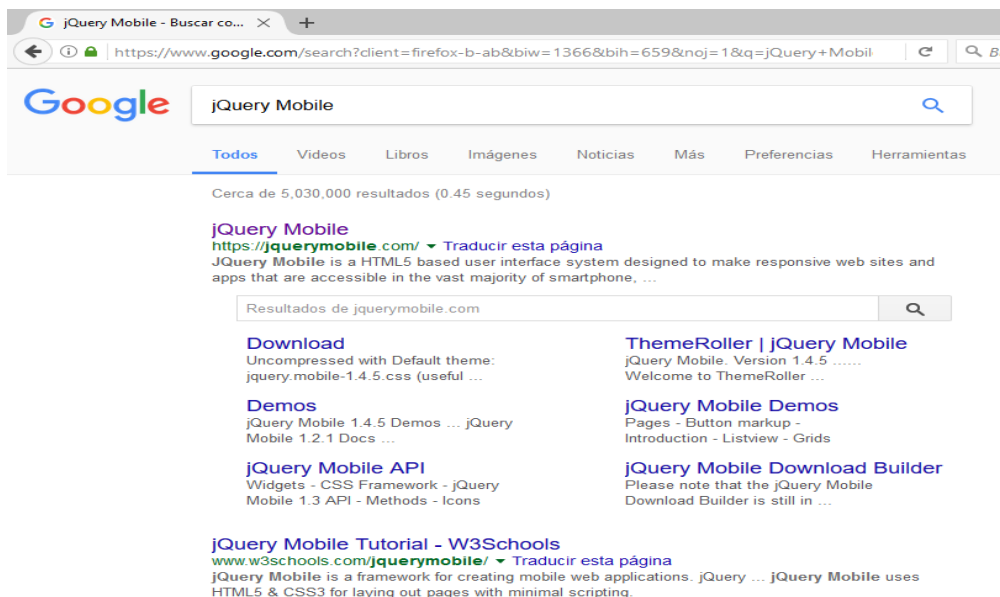
- **Desarrollo multiplataforma:** Android, iOS.
- **Edad de software:** 2004
- **Costo de software:** gratuito
- **Open source:** si
- **Documentación:** <http://www.reactnative.com/>
- **Popularidad:**





## jQuery Mobile

- **Desarrollo multiplataforma:** Android, Windows Phone, Blackberry.
- **Edad de software:** 16 de octubre de 2010.
- **Costo de software:** gratuito.
- **Open source:** si
- **Documentación:** <https://jquerymobile.com/>
- **Popularidad:**

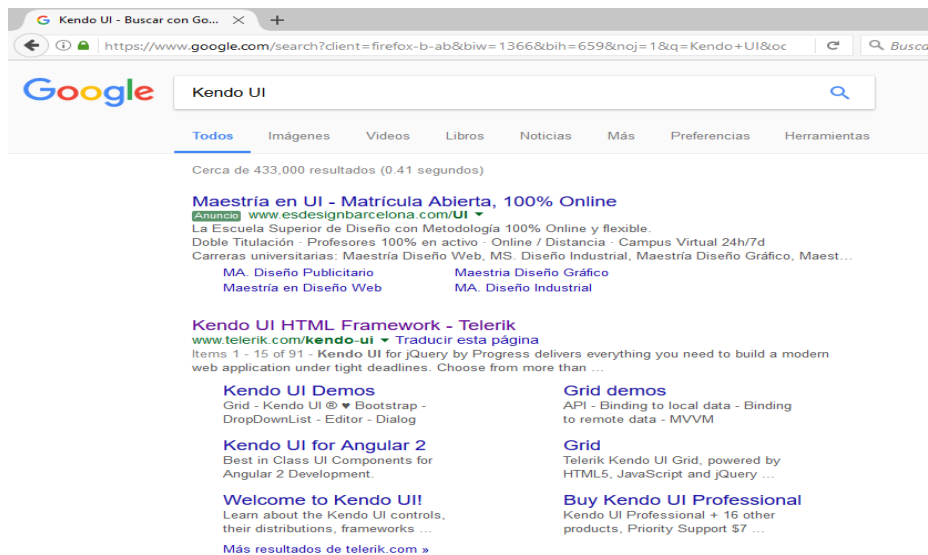


## Kendo UI

- **Desarrollo multiplataforma:** SI, Windows, windows Server, OS X, Android, iOS, Telefono Windows, Chrome para móviles (<http://docs.telerik.com/kendo-ui/intro/supporting/browser-support>)



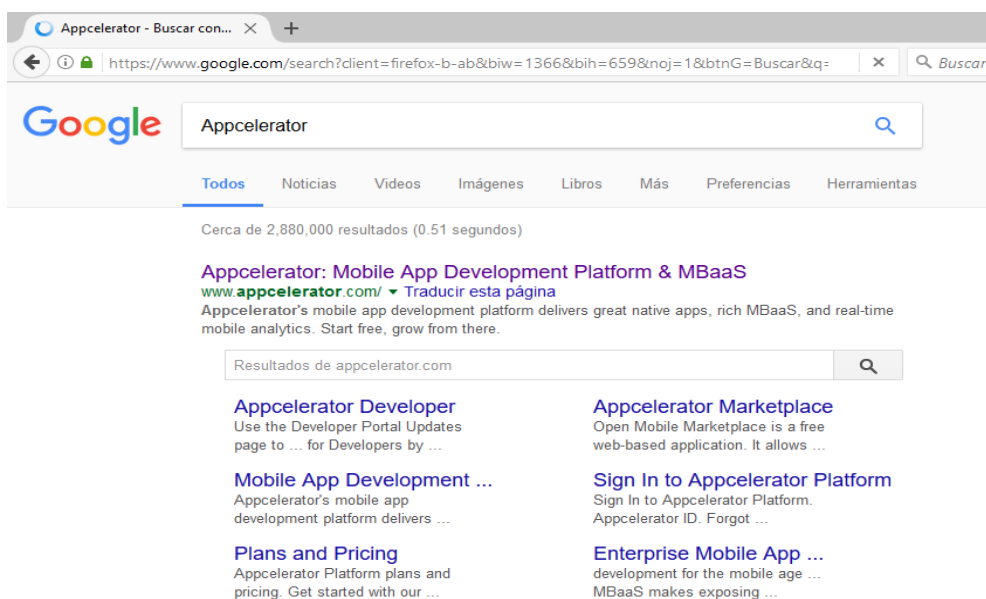
- **Edad de software:** 2002
- **Costo de software:** versión gratuita
- **Open source:** limitado
- **Documentación:** <http://www.telerik.com/kendo-ui>
- **Popularidad:**



## Appcelerator Titanium

- **Desarrollo multiplataforma:** si
- **Edad de software:** 20 de diciembre de 2013
- **Costo de software:** versión gratuita
- **Open source:** si
- **Documentación:** <http://www.appcelerator.com/>
- **Popularidad:**





## ANEXO 2: Historias de usuario

<b>HU:</b> gestión de información usuario	<b>ID:</b> HU2
<b>Prioridad:</b> 4	
<b>Descripción:</b>	El usuario restaurante puede editar información.
<b>Validación:</b>	La aplicación debe mostrar los datos actualizados

<b>HU:</b> Registrar platillo	<b>ID:</b> HU3
<b>Prioridad:</b> 5	
<b>Descripción:</b>	Permite ingresar nombre, descripción, precio y stock del platillo.
<b>Validación:</b>	Validar que el platillo no esté registrado en la base de datos.

<b>HU:</b> editar platillo	<b>ID:</b> HU4
<b>Prioridad:</b> 5	
<b>Descripción:</b>	Permite editar datos como descripción, precio y stock.
<b>Validación:</b>	Mostrar los datos actuales del platillo

<b>HU:</b> eliminar platillo	<b>ID:</b> HU5
<b>Prioridad:</b> 5	
<b>Descripción:</b>	Permite seleccionar un platillo y eliminar.
<b>Validación:</b>	Al seleccionar eliminar, la app debe actualizar el listado de platillos.



<b>HU:</b> listar platillo	<b>ID:</b> HU6
<b>Prioridad:</b> 5	
<b>Descripción:</b>	Permite listar platillos ingresados en la base de datos
<b>Validación:</b>	Los platillos no deben repetirse en el listado

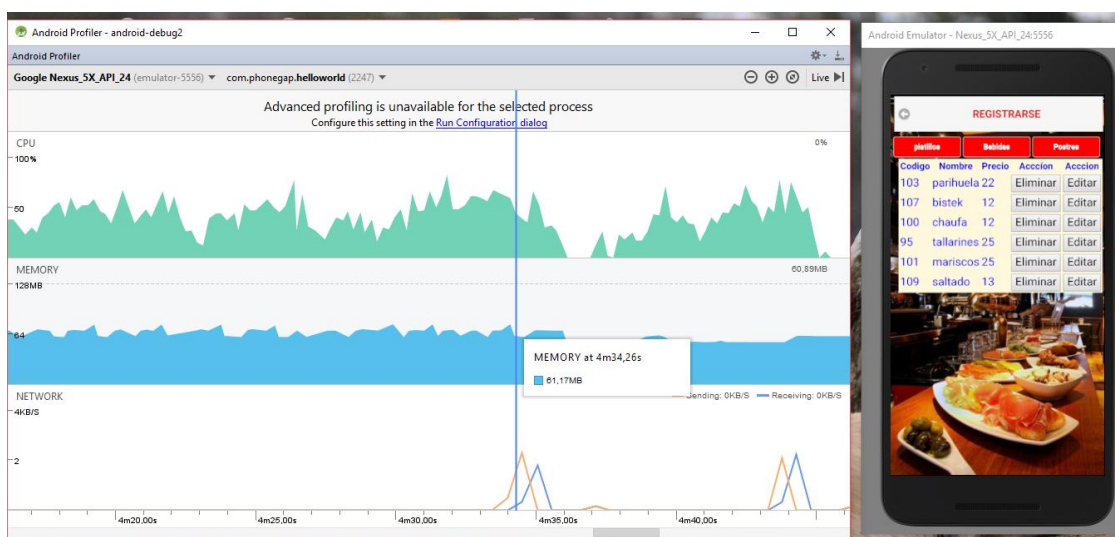
**ANEXO 3: Historias técnicas**

<b>HT:</b> Diseño de base de datos	<b>ID:</b> HT2
<b>Prioridad:</b> 5	
<b>Descripción:</b>	definir la base de datos con la que se desarrollara el aplicativo

<b>HT:</b> Selección de herramientas	<b>ID:</b> HT3
<b>Prioridad:</b> 5	
<b>Descripción:</b>	definir la base de datos con la que se desarrollara el aplicativo

**ANEXO 4: captura de pantalla de evaluación**

En la siguiente ilustración, generada por Android Profiler, corresponde al uso de CPU, memoria y Network por la aplicación móvil, durante el periodo de tiempo que tardó el monitoreo. Donde nos muestra el porcentaje de CPU utilizada, contra el eje de línea de tiempo que representa el tiempo transcurrido en segundos. La curva verde describe la utilización de CPU y la curva celeste describe la utilización de memoria que hizo la aplicación.



ANEXO 5: Diseño de mockups de la app móvil

