



**FACULTAD DE INGENIERÍA,
ARQUITECTURA Y URBANISMO**

ESCUELA ACADÉMICO PROFESIONAL

DE INGENIERÍA DE SISTEMAS

TRABAJO DE INVESTIGACIÓN

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

**PARA OPTAR EL GRADO DE BACHILLER EN INGENIERÍA
DE SISTEMAS**

Autor

De La Cruz Cerón, Roberto Carlos

Asesor

Dr. Mario Fernando Ramos Moscol

Línea de Investigación

Ingeniería de Software

Lima – Perú

2017

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

Aprobación del trabajo de investigación

Apellidos y Nombres: De la Cruz Cerón Roberto Carlos

Autor

Grado/ Apellidos y Nombres: Ing. Miguel Ángel Vidaurre Flores

Asesor Metodológico

Grado/ Apellidos y Nombres:.....

Asesor Especialista

Grado/ Apellidos y Nombres:.....

Presidente de Jurado

Grado/ Apellidos y Nombres:.....

Secretario(a) de Jurado

Grado/ Apellidos y Nombres:.....

Vocal/Asesor de Jurado

Apellidos y Nombres:.....

Autor

Dedicatoria

Este trabajo va dedicado con todo mi cariño a mis padres, quiénes me dieron la educación y la formación necesaria para convertirme en la persona que soy, aunque que sé que jamás podré recompensar todo lo que hicieron por mí, para ellos va este trabajo fruto que tanto esfuerzo me ha costado.

Agradecimiento

En primer lugar agradecer a Dios por haberme dado la vida y la salud para poder llegar a terminar la carrera.

Igualmente mi agradecimiento a mi tutor por su asesoría durante el desarrollo de toda la investigación.

A cada uno de los profesores de la escuela profesional de Ingeniería de Sistemas de la Universidad Señor de Sipán por brindarme los conocimientos y formarme como profesional.

ÍNDICE

INTRODUCCIÓN

CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN	12
1.1. Situación problemática.....	12
1.2. Formulación del problema	17
1.4. Justificación e importancia	18
1.6. Objetivos de la investigación	20
CAPÍTULO II: MARCO TEÓRICO	21
2.1. Antecedentes de la Investigación	21
2.2. Estado del Arte.....	28
2.3. Bases teórico científicas	30
2.3.1. Frameworks	30
2.3.2. Aplicaciones Móviles	35
2.4. Definición de la terminología	45
CAPÍTULO III: MARCO METODOLÓGICO	47
3.1. Tipo y diseño de la investigación	47
3.2. Población y Muestra	47
3.3. Hipótesis	48
3.4. Operacionalización	48
3.5. Métodos, técnicas e instrumentos de recolección de datos	50
3.6. Procedimiento para la recolección de datos	52
3.7. Análisis estadístico e interpretación de los datos.....	55
3.8. Criterios éticos	56
3.9. Criterios de rigor científico	57
CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE RESULTADOS	57
4.1. Resultados en tablas y gráficos	57
4.2. Discusión de resultados.....	61
CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN	66
La selección de proyectos	72

La selección de plataformas	73
La planificación del desarrollo	73
Análisis exploratorio de la planificación	74
Ejemplo de Uso	75
Descripción del proyecto seleccionado	75
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES	81
6.1. Conclusiones	81
6.2. Recomendaciones	82
BIBLIOGRAFÍA	83

Resumen

La presente investigación de título Análisis comparativo entre frameworks de desarrollo para aplicaciones móviles híbridas busca solucionar el problema al momento de elegir el framework adecuado para desarrollar una aplicación móvil híbrida, muchas veces podemos elegir un marco de trabajo sin saber antes si es o no el más adecuado antes de empezar con el desarrollo del proyecto. Esta investigación es del tipo Cuantitativa ya que se tendrán en cuenta índices que nos retornarán números los cuáles nos van a permitir medir el rendimiento y mejor desempeño por cada una de nuestras muestras a analizar. A nivel de tipo de diseño de nuestra investigación ésta es del tipo descriptiva, ya que al tener nuestros resultados obtenidos podremos describir las tecnologías que se van analizar y poder tener en cuentas sus ventajas y desventajas para ser utilizadas en cualquier

proyecto de desarrollo de una aplicación móvil híbrida. Cómo determinar cuál es el framework que mejor se adapta a nuestras necesidades de desarrollo es lo que pretendemos solucionar, entonces para esto se plantea un análisis comparativo donde se realizará una aplicación utilizando ambas tecnologías y podremos medir en base a nuestros indicadores cuál de los dos el más óptimo antes de empezar a desarrollar una aplicación móvil híbrida. Para concluir, ninguna de las dos tecnologías es mejor que la otra sino que ambas tienen características propias, tanto sus ventajas y desventajas que se pueden adaptar a diferentes proyectos, según los requerimientos de éstos.

Palabras clave:

Aplicación híbrida, framework, desarrollo, análisis comparativo, aplicación móvil.

Summary

The present title research Comparative analysis between development frameworks for hybrid mobile applications seeks to solve the problem when choosing the right framework to develop a hybrid mobile application, many times we can choose a framework without knowing before whether or not it is the most adapting before starting with the development of the project. This research is of the quantitative type since we will take into account indexes that will return us numbers which will allow us to measure the performance and better performance for each of our samples to analyze. At the level of the type of design of our research this is of the descriptive type, since to have our obtained results we will be able to describe the technologies that are going to be analyzed and to be able to have in account its advantages and disadvantages to be used in any project of development of an application hybrid mobile phone. How to determine which framework is best suited to our development needs is what we intend to solve, so for this we propose a

comparative analysis where an application will be made using both technologies and we will be able to measure based on our indicators which of the two optimum before starting to develop a hybrid mobile application. In conclusion, neither technology is better than the other, but both have their own characteristics, both their advantages and disadvantages that can be adapted to different projects, according to their requirements.

Keywords:

Hybrid application, framework, development, comparative analysis, mobile application.

INTRODUCCIÓN

En la actualidad el desarrollo de software cada día se hace más necesario dentro de las diferentes empresas y organizaciones tanto públicas como privadas, el tener una aplicación o página web, es tener contacto con el resto del mundo y estar a la vanguardia de la tecnología, ahora con la aparición de los dispositivos móviles se hace necesario que las aplicaciones cumplan ciertos requisitos para que puedan ser no solo visualizados sino también puedan rendir de manera óptima en cara al cliente y para beneficio de las instituciones que hace uso de esta aplicación.

En el mundo de la tecnología móvil actualmente en el mercado hay ciertos sistemas operativos que son los más populares o los más utilizados para determinados dispositivos y cada uno cuenta con su propio lenguaje de desarrollo así como su propia plataforma, he allí donde radica el problema ya que para que una aplicación desarrollada en un lenguaje nativo sólo podrá rendir en la plataforma para la cual ha sido desarrollada, es a raíz de esta problemática es donde aparece el desarrollo de aplicaciones híbridas, ya que nos permite desarrollar aplicaciones móviles indistintamente de una plataforma específica, sino que puede funcionar para cualquier dispositivo.

En la presente investigación se realizó el análisis comparativo entre dos frameworks que nos permiten desarrollar aplicaciones móviles híbridas, para lo cual se desarrolló una aplicación en la que se utilizó ambas

tecnologías con el fin de poder saber las ventajas y desventajas de cada una, se tuvo en cuenta diversos indicadores que nos permitieron medir cada uno de estos frameworks.

Se llegó a la conclusión que ninguno es mejor que el otro, ambos tienen sus propias características y se pueden usar para determinados desarrollos con el fin de poder sacarle el máximo provecho al framework, reduciendo costos y maximizando los recursos en tiempo de desarrollo y entrega del producto.

CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN

1.1. Situación problemática

Colombia

Los Frameworks en la construcción de aplicaciones Web es un tema de reciente desarrollo e investigación en el área del software, y está muy relacionado con el concepto “Web 2.0”, los sistemas modernos de información. Este artículo presenta avances del proyecto que plantea el diseño y creación de un Framework web robusto y eficiente, implementado con herramientas de software libre, que facilite el desarrollo de aplicaciones web corporativas a través de asistentes, con su estructura general (sesiones, seguridad y navegación) y las diferentes páginas de gestión, a partir del modelo de datos de la aplicación.

El Framework Web dispone de las siguientes funcionalidades: Proceso de autenticación, administración de roles y de usuarios, creación y administración de formularios, generación de formularios a partir de tablas, manejo de listas de valores y gestión de registros. Debido a la naturaleza del producto desarrollado, se realizó la selección de un modelo pequeño de dominio para solucionar mediante la herramienta y de esta forma lograr explorar y comprobar que se habían alcanzado

los requerimientos. Fue evidente la importancia de definir estándares de codificación y de administración del código fuente para lograr el nivel de calidad requerido en la implementación del Framework. La comunicación entre la base de datos y el usuario en cuanto a entradas y salidas es facilitada por medio de una interfaz web apoyada en una arquitectura Cliente/Servidor. Es necesario efectuar la revisión de ciertos detalles de compatibilidad entre las partes principales del sistema, las diferentes librerías que permiten realizar AJAX de una manera Cross-browser, manejo de gestores de plantillas, y acceso dinámico a la Metadata de las tablas en diferentes motores de Base de Datos. Aunque se realizaron pruebas de las funcionalidades del sistema, es recomendable aplicar otros modelos de prueba y continuar investigando para el proceso de evolución de este producto software (Martínez, Camacho & Gutierrez, 2010).

Chiclayo

Las organizaciones como las empresas conforme pasa el tiempo van sufriendo diversos cambios, unos son leves y otros son drásticos, incluso, algunos, tienden a transformarlas y fortalecerlas. Las tecnologías han logrado posicionarse de manera firme y decisiva en las estructuras orgánicas de las organizaciones, pero muchas de ellas, aún, son reacias o esquivas a la hora de relacionarse o vincularse

con ellas, trabajan de forma separadas, llevan la información de manera tradicional, utilizan excesivo tiempo: hora/hombres. Los principales perjudicados de este divorcio entre empresa y tecnología son los clientes o usuarios. Por estas razones las organizaciones han empezado a ser más amigables con las tecnologías, sobre todo con las Tecnologías de la Información y Comunicación; varias empresas han empezado a diseñar e implementar Arquitectura Empresarial y darle un nuevo giro al uso de la tecnología en el negocio de su empresa, toman nuevas decisiones para ser competitivos, crear servicios de calidad, satisfacer a los clientes e incrementar la rentabilidad. A través de uso de la Arquitectura Empresarial se está unificando e integrando datos, información, principios, visión, objetivos, estructura orgánica, procesos y todo aquello que beneficia de manera directa o indirecta a la empresa. La arquitectura empresarial es un conjunto coherente de principios, métodos y modelos que se utilizan en el diseño y la realización a nivel empresarial de la estructura organizacional, los procesos de negocio, los sistemas de información y la infraestructura. Una Arquitectura Empresarial explica cómo todos los elementos de las tecnologías de la información en una organización, los procesos, los sistemas, la estructura organizacional y las personas se integran y trabajan de forma conjunta

como un todo ellos. Arquitectura de aplicación bases para cada uno de los sistemas y su relación con el negocio. Y la Arquitectura de tecnología que se basa en la estructura de software y hardware incluyendo área de comunicaciones y soporte. La Arquitectura Empresarial ayuda en el cumplimiento de los objetivos estratégicos a mediano plazo. Asegurando que las metas propuestas en la arquitectura cumplan los requerimientos establecidos obteniendo disminución de costos de TI, y mejoras en la toma de decisiones, logrando integración de los esfuerzos de la organización. El objetivo de la presente investigación es diseñar arquitectura empresarial para mejorar el alineamiento de TI a los objetivos del negocio en la Empresa de Transportes Chiclayo S.A. para lo cual seleccionó Seleccionar un framework que se adapta a las características del rubro del negocio, se analizó la situación actual del negocio, se elaboró una Arquitectura Empresarial AS-IS y Arquitectura Empresarial TO-BE. (Ordoñez, 2017).

Lima

Hoy en día los científicos necesitan de aplicaciones de cómputo para realizar cálculos, validar modelos matemáticos o hacer simulaciones; requiriendo para esto del procesamiento de un gran volumen de datos.

En ocasiones, los requerimientos computacionales para realizar estas tareas exceden las prestaciones que les puede brindar una computadora personal. En vista de esto, es necesario contar con una infraestructura de mayor escala, la cual permita realizar cálculos intensivos

Hay muchas alternativas para implementar la infraestructura necesaria. Por un lado, se puede adquirir una supercomputadora (con altas prestaciones de memoria, procesador) pero a un alto costo. El otro camino viable es construir un sistema de Grid o un Cluster a partir de hardware de bajo costo y que se encuentre ampliamente disponible. Escoger una u otra opción depende de las necesidades particulares de la organización, así como del presupuesto disponible. Si se opta por la segunda opción, existe software open-source para gestionar los recursos del sistema, programar tareas computacionales y ejecutarlas. Un ejemplo de este tipo de sistemas middleware es Berkeley Open Infrastructure for Network Computing (BOINC) desarrollado para dar soporte al cómputo voluntario siendo ampliamente usado en el mundo. La solución propuesta es desarrollar un framework para la generación de interfaces web para proyectos de cómputo intensivo de manera dinámica: que sea a la vez un framework que permita generarlas

(mediante modelado gráfico y textual) y también la plataforma que las hospedará, proveyendo herramientas anexas para propiciar un entorno colaborativo para los investigadores, y que permita mantener a los usuarios informados sobre el estado del sistema (tanto administradores – que estarán al tanto de un estado global, como a los usuarios finales – que estarán pendientes de las tareas que han enviado ellos o sus colaboradores registrados en el mismo proyecto). (Fonseca, 2014).

1.2. Formulación del problema

¿Cómo elegir el framework que mejor se adapte a nuestras necesidades a la hora de implementar una aplicación móvil híbrida?

1.3. Delimitación de a investigación

La investigación tiene como rango de estudio a nivel de dos Frameworks o tecnologías que son Ionic y Phonegap, ambos son frameworks de desarrollo de aplicaciones móviles híbridas que nos permiten desarrollar una aplicación tan sólo utilizando HTML 5, CSS y lenguaje JavaScript, la aplicación desarrollada deberá funcionar de manera óptima en cualquiera de los dispositivos móviles independientemente de su plataforma, para el caso de estudio en la investigación se llevarán a cabo el análisis comparativo para determinar cuál de los dos es el más recomendable para un tipo de

proyecto de desarrollo de aplicación móvil híbrida según se requiera, a fin de tener los resultados esperados para poder explotar al máximo la herramienta adecuada para el desarrollo de software adecuado.

1.4. Justificación e importancia

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

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

Al conocer los frameworks y estar convencido de cuál es el más adecuado para nuestro desarrollo, aporta al ámbito social ya que se

podrá desarrollar manuales de usuario, utilitarios, etc. además se dará a conocer los resultados obtenidos para próximas investigaciones, así como también se aportará de manera científica al tener una idea más clara en cuanto a medición y eficiencia de cada uno de los frameworks para su posterior uso.

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

Este análisis comparativo entre los frameworks Ionic y PhoneGap para el desarrollo de aplicaciones móviles híbridas nos ayudará a poder comprender el por qué cada vez más las aplicaciones móviles híbridas están desplazando a las aplicaciones desarrolladas para plataformas específicas.

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

1.5. Limitaciones de la investigación

No se cuenta con limitaciones en la presente investigación.

1.6. Objetivos de la investigación

Objetivo general

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

Objetivos específicos

- a. Analizar los frameworks existentes en el mercado.
- b. Seleccionar los frameworks para realizar la comparativa.
- c. Implementar una aplicación basada en los frameworks seleccionados.
- d. Evaluar los resultados.

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes de la Investigación

España

Díaz, J. (2014) en su proyecto final de Master con título “Estudio comparativo de diferentes frameworks de alto nivel para el desarrollo de aplicaciones de móviles en Android” se refiere al tema del desarrollo de aplicaciones móviles hechas en Android mediante el uso de frameworks multiplataforma, es decir que tienen la capacidad de generar múltiples copias de una aplicación, compatible con cada una de ellas mediante un solo desarrollo. Cada uno de ellos tiene la capacidad y herramientas para crear una aplicación compatible en Android, como también desarrollar aplicaciones en iOS, Windows phone etc. En el mercado de hoy en día las aplicaciones móviles son esenciales para todo tipo de usos en cualquier entorno, debido a eso, se han hecho tan populares. En el mercado hay una gran competencia por ser la marca más popular entre varias empresas que trabajan en este ámbito de las aplicaciones móviles, como son Google (Android), Apple (iOS), Microsoft (Windows phone) etc. 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 centra 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 comparan 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.

Relevancia

En la investigación se analiza las aplicaciones desarrolladas en Android utilizando diversos frameworks multiplataforma, con el fin de que el desarrollo en Android se realice con el menor esfuerzo, sea flexible y compatible con distintas plataformas.

Ecuador

Mendieta, M. (2014) en su tesis de título “Propuestas de framework de arquitectura empresarial para PYMES basado en un análisis comparativo de los frameworks de zachman y togaf”, considera la dificultad que existe cuando es necesario aplicar un framework en una

empresa tomando en consideración las áreas de negocio, para cubrir todas las necesidades derivadas del entorno global y las soluciones que las áreas tecnológicas han desarrollado para su correcta integración. La investigación expone los conceptos fundamentales necesarios para comprender la terminología de la arquitectura empresarial y frameworks que intervienen, además de exponer el análisis a profundidad de dos frameworks específicos Zachman y TOGAF, con la finalidad de encontrar las herramientas adecuadas para determinar las ventajas y desventajas de cada uno y tener las bases suficientes para determinar fortalezas, oportunidades, debilidades y amenazas de cada uno de ellos. Expone también conceptos necesarios para comprender los alcances de las PYMES en general y la conformación de las PYMES en el Ecuador, además de un análisis FODA tomando en consideración la aplicación de un framework en una PYME. En su último capítulo expone el desarrollo de un framework híbrido en base a la estructura de los frameworks analizados, específicamente en TOGAF y Zachman, con el fin de determinar un método adecuado, facilitando un procedimiento para poder ser aplicado en una PYME.

Relevancia

La investigación nos brinda un alcance sobre la comparación entre dos frameworks de desarrollo para determinar cuál de ellos es el que mejor

se adapta a las necesidades de las PYMES, posteriormente en base a los resultados obtenidos en la comparativa se desarrolla un framework híbrido para ser aplicado en una PYME.

Ecuador

Suárez, E. (2011) en su investigación “Análisis comparativo de los Frameworks EJB3 y ADO.NET Entity framework para el desarrollo de aplicaciones empresariales”, cuyo objetivo establece un conjunto de criterios que permiten evaluar frameworks de desarrollo de la capa de persistencia de aplicaciones empresariales en ambientes Java y .Net, en base a estos criterios se valoró los frameworks Entity Framework y EJB 3 y se identificó fortalezas y debilidades de cada uno de ellos tanto a nivel teórico como práctico a través del desarrollo de prototipos. Los resultados obtenidos permitirán seleccionar el framework más idóneo considerando las necesidades específicas que una organización tenga para desarrollar aplicaciones empresariales.

Obtuvo como resultados que existe una diferencia marcada entre los frameworks evaluados por lo que ambos serían aptos para la creación de una aplicación empresarial, las características técnicas en los ambientes de cada framework son similares.

Concluye que el documento guía obtenido del análisis de los frameworks del presente trabajo permite discriminar la aplicabilidad de

los frameworks EJB3 y ADO.NET Entity Framework para el desarrollo de aplicaciones empresariales.

De acuerdo al análisis de resultados de la evaluación se pudo determinar que Entity Framework es el que ofrece mayores facilidades de uso al crear una capa de persistencia ya que permite la creación y configuración de la mayoría de sus elementos de forma gráfica haciendo de éste un desarrollo más transparente que abstrae los detalles técnicos del framework.

Por el contrario EJB3/JPA es el framework más sólido para una arquitectura más compleja gracias al respaldo de un conjunto amplio de especificaciones como JEE y EJB para la interacción de la capa de persistencia con el resto de capas de una aplicación empresarial. Finalmente se recomienda extender el análisis de las herramientas que le permiten a estos frameworks de persistencia comunicarse con las otras capas con el objetivo de cubrir todo el espectro de una aplicación empresarial.

Relevancia

En la investigación permite ver las diferencias entre uno y otro framework de persistencia para desarrollar aplicaciones empresariales en Java de tal manera que nos da un gran aporte y nos muestra una serie de pasos para poder llegar a un análisis comparativo asertivo.

Piura

Ruiz, J. (2011) en su investigación “Comparativa entre el desarrollo web usando el framework Jboss Seam y el desarrollo tradicional” cuyo objetivo es comparar los costos y los tiempos del desarrollo de manera tradicional y del desarrollo con el framework, así como definir las consideraciones a tener en cuenta antes de optar por usarlo para el desarrollo de una aplicación. Obtuvo como resultados que se encontraron las ventajas y casos particulares en que conviene inclinarse por Seam. Concluye que Seam hace que el desarrollo y mantenimiento de las aplicaciones sea más rápido y fácil. Pero antes de comenzar con un proyecto en Seam, es importante tener en cuenta que la infraestructura necesaria siempre será más cara que la que se usa normalmente para desarrollo tradicional. Finalmente recomienda que si no se cuenta con conocimientos de Seam y la aplicación que se quiere desarrollar es de baja escala, teniendo en cuenta que el periodo de aprendizaje estimado es de 2 meses, es necesario evaluar si el tiempo que tomaría desarrollarla sin un framework podría ser menor al tiempo que tomará aprender y dominar Seam para luego poder programar la aplicación además, para el despliegue en producción de aplicaciones sobre JBoss, ya sea contratando un servicio de hosting o invirtiendo en infraestructura propia, el costo siempre resulta mayor

que el requerido para una infraestructura tradicional, por lo tanto es muy importante tener en cuenta, antes de optar por Seam, la inversión que conlleva ir por esta opción.

Relevancia

Esta investigación nos permite tener un panorama más real sobre el análisis comparativo a realizar, teniendo en cuenta la forma minuciosa de la investigación en mención.

Chiclayo

Sánchez, C. (2015) en su investigación “Análisis comparativo de frameworks para el desarrollo de aplicaciones web en java” cuyo objetivo fue realizar un análisis comparativo de frameworks en Java para el desarrollo de aplicaciones web, aplicando una matriz de un modelo de calidad. Asimismo para el análisis teórico de frameworks se basó en el método QSOS, concebido para definir características comparativas del software de código libre, de esta manera se desarrolló un caso práctico de un sistema de votación para conocer los framework evaluados en su codificación en la práctica. Además se utilizó un modelo de evaluación basado en el Modelo de Construcción de Calidad Individual (IQMC) y las características de calidad propuestas en la norma ISO 2501. Obtuvo como resultados una matriz para la evaluación comparativa para los Frameworks seleccionados.

Los resultados obtenidos mostraron tanto en funcionalidad, fiabilidad, usabilidad, rendimiento, mantenimiento y portabilidad de cada framework que permitirá al arquitecto de software determinar cuál es el frameworks más conveniente para el desarrollo de aplicaciones web. Concluye que no existe una diferencia amplia entre los frameworks comparados, entonces según el proyecto a implementar, el arquitecto de software verá que framework va utilizar. Finalmente se recomienda que antes de elegir un framework a utilizar se deberá tener en cuenta los conocimientos de cada una de las personas que van interactuar con el framework durante el desarrollo del proyecto.

Relevancia

La investigación en mención nos permite ver con claridad la mínima diferencia entre los frameworks de desarrollo para aplicaciones en Java y a tener en cuenta antes de iniciar un proyecto los conocimientos con los que el personal cuenta a la hora de desarrollar sobre una nueva tecnología, esto lo determinará el arquitecto de software encargado del proyecto.

2.2. Estado del Arte

Thomas, P. y Galdamez, N., (2016) en su investigación “Desarrollo y Análisis de Rendimiento de Aplicaciones Multiplataforma” nos

presenta una línea de investigación y desarrollo, que tiene por objeto estudiar temas relacionados con aspectos de Ingeniería de Software, orientados al desarrollo de aplicaciones móviles multiplataforma.

Los autores nos dicen que la computación móvil puede definirse como un entorno de cómputo con movilidad física. Un usuario debe ser capaz de acceder a datos, información u otros objetos lógicos desde cualquier dispositivo en cualquier red, al mismo tiempo que va moviéndose, cambiando su locación geográfica. El desarrollo de aplicaciones para dispositivos móviles plantea nuevos desafíos originados en las características únicas de esta actividad. La necesidad de tratar con diversos estándares, protocolos y tecnologías de red; las capacidades limitadas, aunque en constante crecimiento, de los dispositivos; las restricciones de tiempo impuestas por un mercado altamente dinámico y la existencia de distintas plataformas de hardware y software son sólo algunas de las dificultades a las que se enfrentan los desarrolladores en esta área. Las aplicaciones se generan en un entorno dinámico e incierto. En su mayoría se trata de aplicaciones pequeñas, no críticas, destinadas a un gran número de usuarios finales que son liberadas en versiones rápidas para poder satisfacer las demandas del mercado. Todas las particularidades previamente mencionadas hacen que el desarrollo de software para

dispositivos móviles difiera considerablemente del tradicional. Ello conduce a nuevas prácticas y metodologías que promueven el crecimiento de la Ingeniería de Software como disciplina, acompañando este proceso de desarrollo tecnológico.

Finalmente se tiene como los resultados obtenidos avanzar en la capacitación continua de los miembros involucrados en esta línea de investigación así como también utilizar y comparar el uso de diversos frameworks para el desarrollo de aplicaciones móviles nativas y multiplataforma.

2.3. Bases teórico científicas

En este apartado se dará a conocer algunos conceptos de vital relevancia para esta investigación, con la finalidad de situar al problema dentro de un conjunto de conocimientos.

2.3.1. Frameworks

Framework es un término que se utiliza a menudo en la computación general, básicamente para referirse a un conjunto de bibliotecas utilizadas para implementar la estructura estándar de una aplicación, es decir, crear el cascarón de la aplicación. Todo esto con la finalidad de ahorrar líneas de código, de esta manera facilitando el desarrollo

ya que se reutiliza el código para que el desarrollador no tenga que estar programando lo mismo para cada aplicación.

Gamma, E. (1995) nos dice que un Framework determina la arquitectura de la aplicación. Éste es un buen enfoque ya que el framework se encarga de definir la estructura general, sus particiones en clases y objetos, las responsabilidades clave así como la colaboración entre dichas clases y objetos.

Gamma, E. (1995) nos explica que un framework captura las decisiones de diseño que son comunes a su dominio de aplicación, un framework no sólo promueve la reutilización de código sino también el diseño.

Un framework nos ayuda a que se desarrolle una aplicación de una manera mucho más sencilla y rápida ya que no se pierde el tiempo en algunos detalles de diseño que muchas veces quitan más tiempo del que tomó construir en sí la lógica de la aplicación. Además de que las aplicaciones que se construyen tienen estructuras similares, más fáciles de mantener y consistentes para los usuarios. Cabe mencionar que esto tiene como consecuencia una mínima pérdida de libertad en temas de diseño.

En algunas ocasiones el desarrollador tiene ciertas dificultades para diseñar una aplicación, esto es todavía mucho más difícil para el desarrollador del framework, ya que al desarrollar en un framework requiere de ciertos cuidados, porque cuándo se lanza uno nuevo todos esperamos que pueda servir para muchos tipos de aplicaciones pero con arquitectura y requerimiento similar. Es decir trata de englobar toda una gama de aplicaciones dentro de un mismo estándar, lo cuál puede ser un éxito o un fracaso, por tal motivo se intentan crear frameworks lo más extensibles y flexibles posibles, para que con algunos cambios se puedan corregir de manera rápida y sencilla. Las aplicaciones que se desarrollan a partir de un Framework, está ligada al mismo, por eso las aplicaciones debe evolucionar y crecer al mismo tiempo que crece el framework, ya que un cambio en el mismo significará un cambio notable en la aplicación, dependiendo de qué tan drástico sea el cambio.

Gutierrez, J., (s.f.) nos dice que el concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrírse nos. En general, con el término framework, nos estamos

refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Dentro de los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Por otro lado Recaman, H. y Guerrero, C. (2009) definen un Framework como un conjunto de componentes físicos y lógicos estructurados de manera que permiten ser reutilizados en el diseño y desarrollo de nuevos sistemas de información.

2.3.1.1. PhoneGap

“Es un framework que permite desarrollar aplicaciones móviles para múltiples plataformas, usando los lenguajes de programación HTML5, CSS y JavaScript en conjunto con comandos propios para acceder a funcionalidades del teléfono. Es uno de los frameworks más utilizados en la red con más de 400 mil desarrolladores.”

(PhoneGap, 2013).

El núcleo del motor de PhoneGap es 100% de código abierto, en virtud del proyecto de Apache Cordova. PhoneGap proporciona una interfaz de programación de aplicaciones (API) que permite acceder a múltiples de las funcionalidades nativas del sistema operativo utilizando JavaScript. “El API de PhoneGap se encarga de traducir los comandos JavaScript y realizar la comunicación con el sistema operativo nativo. A pesar de que las aplicaciones desarrolladas en PhoneGap utilizan HTML, CSS y JavaScript, el producto final es un archivo genérico que puede ser traducido a múltiples extensiones para trabajar en diferentes sistemas operativos.” (PhoneGap, 2013).

2.3.1.2. Ionic Framework

Ionic es un Framework de código abierto SDK para desarrollo de aplicaciones móviles híbridas. Construido sobre angularjs y Apache Cordova, Ionic ofrece herramientas y servicios para el desarrollo de aplicaciones móviles híbridos utilizando tecnologías web como CSS, HTML5, y Sass. Las aplicaciones pueden ser construidas con estas tecnologías Web y luego distribuidas a través de tiendas nativas de aplicaciones para instalar en los dispositivos mediante el aprovechamiento de Córdoba. Ionic fue creado por Max Lynch, Ben Sperry, y Adam Bradley de Drifty Co. en 2013, y es utilizado por los desarrolladores de software de todo el mundo (Ionic, 2013).

2.3.2. Aplicaciones Móviles

Es un software de aplicación que es creado para correr en dispositivos móviles, como smartphones y tablets. Las primeras aplicaciones móviles aparecieron tan temprano como en los principios del siglo 20, la mayoría de ellos consistían en juegos de video, calculadoras, y editores de tono de llamada. Un Software más sofisticado comenzó a aparecer en dispositivos móviles en el 2008, aunque la mayoría de ellos fueron creados a conveniencia general e incluían cosas como un calendario, información sobre el clima, o una lista de contactos.

2.3.2.1. Aplicación Nativa

Las aplicaciones nativas son específicamente creadas para una plataforma, como iPhone o Android, y pueden hacer uso de todas las características del dispositivo, como la cámara, GPS, lista de contactos, y notificaciones del sistema. Son llamadas aplicaciones nativas porque son “nativas” de cualquiera que sea el dispositivo para el que fueron desarrolladas para usar. Estas son las aplicaciones que usted generalmente puede encontrar en lugares como Google Play o la Tienda de Aplicaciones de Apple (Apple’s App Store). Estas aplicaciones son generalmente las más rápidas de los tres tipos de aplicaciones.

2.3.2.2. Aplicación Web

Una aplicación web es un sitio web que actúa como una aplicación nativa, pero no están instaladas o implementadas de la misma forma. Ellas usan un navegador web y usualmente usan HTML5. También se apoyan en un navegador web en común de acuerdo para interpretar la aplicación. Sin embargo, la distinción entre aplicaciones nativas y aplicaciones web está volviéndose cada vez más pequeña mientras más sitios comienzan a usar HTML5. Una aplicación web puede tener algunas ventajas sobre una aplicación nativa porque son más fáciles de crear y desarrollar y proveen compatibilidad entre plataformas.

2.3.2.3. Aplicación Híbrida

El enfoque híbrido combina desarrollo nativo con tecnología Web. Usando este enfoque, los desarrolladores escriben gran parte de su aplicación en tecnologías Web para múltiples plataformas, y mantienen el acceso directo a Aplicaciones nativas cuando lo necesitan. La porción nativa de la aplicación emplea APIs de sistemas operativos para crear un motor de búsqueda HTML incorporado que funcione como un puente entre el navegador y las APIs del dispositivo. Este puente permite que la aplicación híbrida aproveche todas las características que ofrecen los dispositivos

modernos. Los desarrolladores de aplicaciones pueden optar por codificar su propio puente o bien aprovechar soluciones ya construidas, como PhoneGap, una biblioteca de fuente abierta que provee una interfaz JavaScript uniforme para funcionalidades de dispositivos seleccionados que son iguales en todos los sistemas operativos. La porción nativa de la aplicación se puede desarrollar independientemente, pero algunas soluciones del mercado ofrecen este tipo de contenedor nativo como parte de su producto, lo que brinda al desarrollador formas de crear una aplicación avanzada que utilice todas las funciones del dispositivo usando únicamente lenguajes Web. En algunos casos, una solución va a permitir que el desarrollador utilice cualquier conocimiento nativo que pueda tener para adaptar el contenedor nativo a las necesidades únicas de la organización. La porción Web de la aplicación puede ser una página Web que resida en un servidor o bien un conjunto de archivos HTML, JavaScript, CSS y medios, incorporados en el código de la aplicación y almacenados localmente en el dispositivo. Ambos enfoques presentan ventajas y desventajas. El código HTML que está alojado en un servidor permite que los desarrolladores introduzcan pequeñas actualizaciones en la aplicación sin tener que seguir el proceso de entrega y aprobación que algunas tiendas de aplicaciones requieren.

A continuación, citamos las ventajas y desventajas del desarrollo de aplicaciones móviles Híbridas.

2.3.2.3.1. Ventajas

- a. Uso de los recursos del dispositivo y del sistema operativo.
- b. El costo de desarrollo puede ser menor que el de una nativa.
- c. Son multiplataforma.
- d. Permite distribución a través de las tiendas de su respectiva plataforma.

2.3.2.3.2. Desventajas

- a. La documentación puede ser un poco escasa y desordenada

Una aplicación híbrida es justo lo que el nombre indica – una combinación entre una aplicación web y una aplicación nativa. Ellas “viven” en una tienda de aplicaciones y pueden hacer uso de las características del dispositivo casi de la misma forma que una aplicación nativa.

Las Aplicaciones híbridas tratan de buscar lo mejor de las Apps

Nativas y las Apps Web para ofrecerlas como Aplicación Móvil.

Explicado de manera sencilla, la aplicación se programa como una App Web (con HTML5, CSS y JS) y luego el framework se encarga de adaptar la vista web a una vista de dispositivo móvil, traduciendo todas las tareas de la vista web a acciones propias del teléfono/tablet.

Con esto, una Aplicación Híbrida puede llegar a múltiples plataformas móviles con un esfuerzo casi nulo, pero al costo de perder acceso a ciertas capacidades de cada dispositivo. (Angulo, R. 2013).

2.3.3. Lenguajes

Para la creación de aplicaciones web se utilizan múltiples lenguajes. En este caso, se han utilizado diferentes lenguajes de programación web como pueden ser HTML, PHP o JavaScript.

2.3.3.1. Hyper Text Markup Language (HTML)

El Hyper Text Markup Language (HTML) [Specification, 2004], es un lenguaje de marcado, diseñado para estructurar textos y definir su presentación en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Mozilla, Firefox, Netscape o Explorer, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos. Contrariamente a otros lenguajes de programación, el HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma con la que deben aparecer el texto, así como las imágenes y los demás elementos, en la pantalla del ordenador.

2.3.3.2. JavaScript

No debe confundirse Java con JavaScript. Java es un lenguaje completo que permite aplicaciones independientes, mientras que JavaScript es un lenguaje de programación interpretado empleado principalmente en el desarrollo de páginas web, siendo similar al lenguaje Java y al lenguaje C, base de la programación en los sistemas operativos, como Windows. JavaScript es un lenguaje de programación creado con el objetivo de integrarse en HTML y facilitar la creación de páginas interactivas sin necesidad de utilizar scripts de cierta nivelación o Java. (Orós, 2002)

Es un lenguaje de programación orientado a objetos, diseñado para el desarrollo de aplicaciones cliente – servidor a través de internet. JavaScript se puede incluir en cualquier documento HTML, o todo aquel que termine traducándose en HTML en el navegador del cliente; ya sea PHP, ASP, entre otros. Incluir código directamente en una estructura HTML es una práctica invasiva, y no recomendada. El método correcto que define la W3C es incluir JavaScript como un archivo externo, tanto por cuestiones de accesibilidad, como practicidad y velocidad en la navegación.

2.3.3.3. PHP

PHP, son siglas de Hypertext Pre - processor, es programación que se ejecuta en el servidor antes de ser enviados a un navegador. Esto permite realizar acciones como consultar bases de datos o dar formato a los resultados. PHP es junto a ASP (Active Server Pages) los dos lenguajes de programación de este tipo más extendidos en Internet. Es de fácil uso, y es muy parecido a los lenguajes más comunes de programación estructurada, como C y Perl, PHP permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas. Debido al diseño de PHP, también es posible crear aplicaciones con una interfaz gráfica para el usuario (también llamada GUI), utilizando la extensión PHP-Qt o PHP-GTK. Y cuenta con la posibilidad de ser usado desde la línea de órdenes, de la misma manera como Perl o Python pueden hacerlo, esta versión de PHP se llama PHP CLI (Command Line Interface). Su ejecución se da en él un servidor web, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución, cuando el cliente hace una petición al servidor para que se le envíe una página web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica,

permitiendo modificar el contenido a enviar, y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente. Además es posible utilizar PHP para generar archivos PDF, Flash, así como imágenes en diferentes formatos.¹ Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de aplicaciones web muy robustas, también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX, Linux o Mac OS X y Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.² El modelo PHP puede ser visto como una alternativa al sistema de Microsoft que utiliza ASP.NET/C#/VB.NET, a ColdFusion de la compañía Adobe (antes Macromedia), a JSP/Java de Sun Microsystems, y al famoso CGI/Perl, aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU.

2.3.3.4. AJAX

El término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de

aplicación web que estaba apareciendo. En realidad, el término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML".

El artículo define AJAX de la siguiente forma: "Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes." Las tecnologías que forman AJAX son:

- a. XHTML y CSS, para crear una presentación basada en estándares.
- b. DOM, para la interacción y manipulación dinámica de la presentación.
- c. XML, XSLT y JSON, para el intercambio y la manipulación de información
- d. XMLHttpRequest, para el intercambio asíncrono de información.
- e. JavaScript, para unir todas las demás tecnologías.

Desarrollar aplicaciones AJAX requiere un conocimiento avanzado de todas y cada una de las tecnologías anteriores.

En las aplicaciones web tradicionales, las acciones del usuario en la página (pinchar en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición

del usuario, el servidor devuelve una nueva página HTML al navegador del usuario.

Esta técnica tradicional para crear aplicaciones web funciona correctamente, pero no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, su uso se convierte en algo molesto. AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

Las peticiones HTTP al servidor se sustituyen por peticiones JavaScript que se realizan al elemento encargado de AJAX. Las peticiones más simples no requieren intervención del servidor, por lo que la respuesta es inmediata. Si la interacción requiere una

respuesta del servidor, la petición se realiza de forma asíncrona mediante AJAX. En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor.

Desde su aparición, se han creado cientos de aplicaciones web basadas en AJAX. En la mayoría de casos, AJAX puede sustituir completamente a otras técnicas como Flash. Además, en el caso de las aplicaciones web más avanzadas, pueden llegar a sustituir a las aplicaciones de escritorio.

2.4. Definición de la terminología

Alta cohesión: Una de las características esperables en el desarrollo de software junto con el bajo acoplamiento y la mayor modularidad. (Diccionario de Tecnología e Informática, s.f.).

Android: desarrollado por Google para teléfonos inteligentes con pantalla táctil. Basado en el kernel de Linux. Es el sistema operativo móvil con más ventas en tablets y teléfonos inteligentes desde 2013.

Aplicación: Una aplicación (también llamada app) es simplemente un programa informático creado para llevar a cabo o facilitar una tarea en un dispositivo informático.

Bajo acoplamiento: Es la medida en que los cambios de un componente tiende a necesitar cambios de otro componente.

Css: Son las siglas de Cascading Style Sheets - Hojas de Estilo en Cascada que es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación.

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

HTML: Hyper Text Mark-up Language. Tecnología de presentación sobre la que se basa el Internet desde sus inicios. Constituye la capa de contenido de una aplicación web ya que en ella se sitúa la información que se desea compartir a través de la web.

Implementación: Es la instalación de una aplicación informática, realización o la ejecución de la misma.

JavaScript: Lenguaje de programación orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Modelo: Es la representación de la información con la cual el sistema opera.

Vista: Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de

usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

CAPÍTULO III: MARCO METODOLÓGICO

3.1. Tipo y diseño de la investigación

3.1.1. Tipo de Investigación

El tipo de investigación según el enfoque abstracto es de tipo investigación aplicada.

3.1.2. Diseño de Investigación

El diseño de la investigación es Cuasi-Experimental.

3.2. Población y Muestra

Se muestra el siguiente cuadro indicando la población y la muestra.

Indicador	Cantidad	Descripción
Población	10	Se encontró un total de 10 frameworks para el desarrollo de las aplicaciones móviles híbridas.
Muestra	2	Se realiza el análisis comparativo de Ionic y Phonegap

3.3. Hipótesis

En esta hipótesis al analizar y comparar ambos frameworks de desarrollo para aplicaciones móviles, se espera que ni uno ni el otro sea el mejor, ya que ambos tienen características propias, así como sus ventajas y desventajas.

Se tiene en cuenta que Phonegap tiene mayor compatibilidad con más plataformas a diferencia de Ionic, pero este último sólo trabaja con AngularJs que al ser un producto de Google tiene una gran cantidad de documentación, información y soporte. Phonegap puede trabajar con cualquier otro Framework JS y CSS.

Se tiene en cuenta que ambos son open source, de tal manera que son de libre distribución, lo cual es una gran ventaja para sea cuál sea el framework que se elegirá para un desarrollo de alguna aplicación móvil.

Con respecto a facilidad de uso, facilidad de configuración y facilidad de administración Ionic tiene esa ventaja, cabe también mencionar que ambos frameworks trabajan sobre Apache Córdoba.

3.4. Operacionalización

Variable	Dimensiones	Indicadores	Técnicas e instrumentos de recolección de datos

Framework	Curva de aprendizaje	de N° de horas para producir el primer módulo	Entrevista
	Proceso de instalación	de Nivel de dificultad en la instalación	Entrevista y observación
	Consumo de memoria	de Consumo de Memoria Ram	Observación
	Recursos	Consumo de %	Observación
		CPU	
	Compatibilidad con plataformas nativas	N° de plataformas compatibles	Entrevista
Aplicaciones Móviles Híbridas	Consumo de recursos	de Consumo de memoria Ram	Observación
		Tiempo de respuesta	Observación

		Consumo de %CPU	Observación
--	--	--------------------	-------------

3.5. Métodos, técnicas e instrumentos de recolección de datos.

3.5.1. Métodos

Se opta por el método teórico, a continuación se describe cada uno de los métodos teóricos utilizados.

3.5.1.1. Análisis; este método se caracteriza por la descomposición del problema de investigación en diversas partes. Es recomendable que estas partes sean técnicamente indivisibles, es decir, que su división no conlleve a un elemento o componente comprensible.

3.5.1.2. Síntesis; consiste en la integración de las partes de un problema y la determinación de la característica en conjunto que estas generan al problema general.

3.5.1.3. Modelación; este método consiste en la elaboración de un modelo que represente de manera exacta el problema en investigación o la propuesta de solución. El propósito de modelar el problema de investigación es comprender su estructura y dinámica, de manera tal que pueda plantearse una alternativa de solución integral, efectiva y sostenible. Cuando es utilizado para el planteamiento de la alternativa de solución, establece con

precisión cada componente de la solución y la forma en que contribuirá a la solución del problema en estudio.

3.5.1.4. Histórico; método que consiste en el análisis del problema de estudio, desde sus primeros indicios hasta la actualidad. Por cada una de las etapas en que se fue desarrollando, el investigador establece las características de su comportamiento y los diversos factores que influyeron en su evolución, hasta el día actual. De esta manera tendrá también información suficiente para hacer las proyecciones de su comportamiento futuro, a través de un proceso prospectivo.

3.5.1.5. Deducción; comparación de información del caso de estudio en relación a la información general de otros casos: investigaciones, casuísticas, teorías, etc. De esta manera se podrá explicar los resultados de nuestra investigación a partir de conocimientos teóricos o experiencias de otras circunstancias. Es por esa razón que se considera a este método como la integración de conocimientos de “lo general” a lo “específico”.

3.5.1.6. Inducción; vienen a ser las conclusiones a partir de información particular. Es decir, hacer inferencias o conclusiones en donde se hace una generalización de los resultados obtenidos en casos

específicos. Por eso se dice que este método tiene una transición de lo particular a lo general.

3.5.2. Técnicas

Se ha empleado técnicas como son las entrevistas y la observación, se detalla a continuación un resumen de cada una.

3.5.2.1. Entrevista; esta técnica tiene como propósito recolectar información de una persona en particular a la vez. Permite entablar una conversación fluida y obtener información más precisa. Su instrumento es el Cuestionario o también la guía de entrevista.

3.5.2.2. Observación; tiene como propósito recolectar información de acontecimientos, hechos, procesos, etc. Es muy utilizada cuando no es conveniente solamente preguntarle a las personas sobre ciertos aspectos, sino también corroborarlo de manera directa.

3.5.3. Instrumentos

Los instrumentos utilizados son el cuestionario de entrevista y la hoja de observaciones que será anexado en el siguiente informe ya que se está trabajando en eso.

3.6. Procedimiento para la recolección de datos

Se presenta el siguiente cuadro con las herramientas utilizadas y su descripción:

N°	Técnica	Instrumento	Fuente/Informante	Descripción
----	---------	-------------	-------------------	-------------

1	Entrevista	Cuestionario	Conocedores de la herramienta	Entrevistar a 4 personas conocedoras de los frameworks a estudiar a fin de tener una idea más clara de lo que ellos piensan sobre
---	------------	--------------	-------------------------------	---

				estas herramientas y cuál ha sido su experiencia con cada una.
--	--	--	--	--

2	Observación	Hoja de observación	Ciclo de procesos	Recolectar información correspondiente al uso de determinados frameworks al momento del desarrollo de aplicaciones móviles híbridas, esto nos dará una visión más clara de cómo funcionan
---	-------------	---------------------	-------------------	---

				nuestras herramientas del tema de estudio, pudiendo identificar sus ventajas y desventajas.
--	--	--	--	--

3.7. Análisis estadístico e interpretación de los datos

Para analizar nuestros datos recogidos se utilizaron las siguientes fórmulas.

3.7.1. La Sumatoria

Es conocida también como operación de suma, notación sigma o símbolo suma, esta fórmula nos permite obtener el resultado de la suma de varios sumandos e incluso infinitos sumandos, se expresa con la letra griega en mayúscula sigma (Σ).

Su fórmula es:

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

3.7.2. La Media

Viene a ser el valor promedio de las muestras y es independiente de las amplitudes de los intervalos. Se simboliza como \bar{x} y se encuentra sólo para variables cuantitativas. Se encuentra sumando todos los valores y dividiendo por el número total de datos.

Su fórmula es:

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

3.8. Criterios éticos

Teniendo en cuenta el Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica se tendrán en cuenta los siguientes criterios:

3.8.1. Ambiente; mostrando un amplio respeto por mantener el medio ambiente se realiza la investigación teniendo en cuenta la utilización de material informático que no perjudica el ambiente ni el deterioro del mismo utilizando tecnología de punta, así como también se fomenta al desarrollo de aplicaciones móviles que puedan colaborar de alguna forma con mantener el medio ambiente.

3.8.2. Objetividad; la situación encontrada será mostrada con un alto grado de imparcialidad, favoreciendo al mismo tiempo con material para posteriores investigaciones y mejoras de la misma.

3.8.3. Originalidad; toda información presentada será citada dejando en claro que no existe plagio intelectual.

3.8.4. Veracidad; la información aquí mostrada es totalmente verdadera cuidando la confidencialidad de la misma.

3.9. Criterios de rigor científico

Se tomarán en cuenta los siguientes criterios de rigor científico:

3.9.1. Confiabilidad; ya que toda la información recopilada es en base a entrevistas y observación en el mismo escenario, esto da soporte a que la información es confiable, precisa e íntegra.

3.9.2. Validación; la información que recogida es validada ya que anterior al recojo de la información se realizó una proyección para saber qué tipo de resultados se obtendrían, de manera que en esta etapa se valida correctamente de acuerdo a nuestras variables de operacionalización.

3.9.3. Fiabilidad; se ha minimizado el error aleatorio.

3.9.4. Replicabilidad; es posible repetir esta investigación.

CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1. Resultados en tablas y gráficos

A continuación se muestra el contenido de los resultados obtenidos según nuestras variables e indicadores al realizar en análisis comparativo entre ambos frameworks estudiados.

4.1.1. Variables e indicadores evaluados

La medición se realizará con los siguientes indicadores

4.1.1.1. Frameworks

- a. **Curva de aprendizaje**; este indicador nos permite poder medir el tiempo en horas de esfuerzo para desarrollar el primer módulo o CRUD de nuestra aplicación, posteriormente los tiempos se van mejorando.

Framework	Horas de esfuerzo	Unidad de medida
Ionic	32	Horas
Phonegap	40	Horas

- b. **Nivel de dificultad durante la instalación**; se muestra un cuadro con los valores que nos permitirán evaluar el grado o nivel de dificultad que demanda la instalación del framework a analizar.

Framework	Nivel de dificultad (1-Fácil, 2-Intermedio, 3-Difícil)
Ionic	1
Phonegap	1

- c. **Consumo de recursos**; con este indicador nos permitirá medir que tanto consume el framework utilizado en el desarrollo, tanto el consumo de memoria RAM así como también el porcentaje de consumo de CPU.

Framework	Consumo de RAM	% de CPU
-----------	----------------	----------

Ionic	68.50 MB	1.8%
Phoneygap	64.20 MB	1.6%

- d. **Plataformas compatibles;** se ha colocado el número de plataformas con las cuales el framework es compatible y nos permitirá mejorar los tiempos de desarrollo y optimizar el uso del mismo.

Framework	N° de plataformas compatibles
Ionic	4
Phoneygap	9

4.1.1.2. Aplicaciones Móviles Híbridas

- a. **Tiempo de respuesta;** se mide el tiempo que tarda en responder nuestra aplicación ya desarrollada, con respecto a alguna acción del CRUD.

Framework	Tiempo de Respuesta			
	Create	Read	Update	Delete
Ionic	1372 ms.	996 ms.	1203 ms.	951 ms.
Phoneygap	1132 ms.	865 ms.	1132 ms.	861 ms.

- b. **Consumo de memoria RAM;** se puede medir el consumo de memoria RAM que demanda nuestra aplicación utilizando un emulador de un dispositivo móvil.

Framework	Consumo de RAM
Ionic	50.45 MB
Phoneygap	60.65 MB

c. **Consumo en porcentaje de CPU;** de la misma forma que en el caso anterior se medirá el consumo en porcentaje de CPU que demanda nuestra aplicación utilizando un emulador de un dispositivo móvil.

Framework	Consumo en % de CPU
Ionic	1.4%
Phoneygap	1.8%

4.1.2. Contrastación con la Hipótesis

En contraste con la hipótesis planteada se llega a demostrar que efectivamente ambos frameworks son muy buenos ambos con sus respectivas ventajas y desventajas, como se mencionó en la hipótesis Phoneygap tiene mucho más compatibilidad con otras plataformas, de tal manera que por esta parte se pone por encima de Ionic, de la misma forma Ionic al trabajar con Angular JS, la curva de aprendizaje sería más alta que la de Phoneygap, además se reduciría el tiempo de desarrollo ya que Ionic de la mano con Angular JS al empezar un nuevo proyecto nos crea desde ya el cascaron implementado con un sistema de plantillas, ahorrando así horas de código.

En el consumo de recursos entre ambos framework la diferencia sería mínima ya que ambos al ser de Apache Córdoba, básicamente consumirían la misma cantidad de recursos.

Al tener las aplicaciones ya desarrolladas y midiendo el tiempo de respuesta se obtiene también que ambos son similares en sus resultados, demostrando más velocidad en tiempo de respuesta la aplicación realizada con Phonegap.

4.2. Discusión de resultados

A partir de los resultados obtenidos aceptamos lo que se asegura en la hipótesis que establece que ambos frameworks son muy buenos ambos con sus propias cualidades, ventajas y desventajas, llegando a ser Phonegap el que es más compatible con plataformas nativas y Ionic es el que tiene la curva de aprendizaje más alta, demostrando que se puede desarrollar una aplicación en menos tiempo y desde cero utilizando Angular JS.

Por otra parte estos resultados guardan relación con lo que dice Suárez, E. (2011) “Se identificaron fortalezas y debilidades de cada uno de ellos tanto a nivel teórico como práctico a través del desarrollo de prototipos”.

De la misma manera nuestros resultados obtenidos se asemejan a lo dicho por Sánchez, C. (2015) que “Concluye que no existe una diferencia amplia entre los frameworks comparados, entonces según el proyecto a

implementar, el arquitecto de software verá que framework va utilizar. Finalmente se recomienda que antes de elegir un framework a utilizar se deberá tener en cuenta los conocimientos de cada una de las personas que van interactuar con el framework durante el desarrollo del proyecto”.

A continuación se muestra un cuadro resumen con todos nuestros indicadores:

VARIABLES	MUESTRA	Curva de aprendizaje (horas primer CRUD)	Nive de dificultad de instalación (1 - fácil, 2 - intermedio, 3 - difícil)	Consumo de memoria RAM	% de consumo CPU	Tiempo de Respuesta CRUD				Número de plataformas compatibles
						C	R	U	D	
FRAMEWORK	IONIC	32	1	68.50 MB	1.8%	x	x	x	x	4
	PHONEGAP	40	1	64.20 MB	1.6%	x	x	x	x	9
APLICACIONES MÓVILES HÍBRIDAS	IONIC	x	x	50.45 MB	1.4%	1372 ms.	996 ms.	1203 ms.	951 ms.	x
	PHONEGAP	x	x	60.65 MB	1.8%	1132 ms.	865 ms.	1132 ms.	861 ms.	x

Resumen del análisis comparativo teniendo en cuenta los Frameworks

FRAMEWORK	PUNTAJE OBTENIDO (5 - GANADOR, 0 - PERDEDOR, 2.5 - EMPATE)					TOTAL	PROMEDIO
	CURVA DE APRENDIZAJE	DIFICULTAD DE INSTALACIÓN	CONSUMO DE MEMORIA RAM	% DE CPU	PLATAFORMAS COMPATIBLES		
IONIC	5	2.5	0	0	0	7.50	1.50
PHONEGAP	0	2.5	5	5	5	17.50	3.50

Resumen del análisis comparativo teniendo en cuenta las Aplicaciones

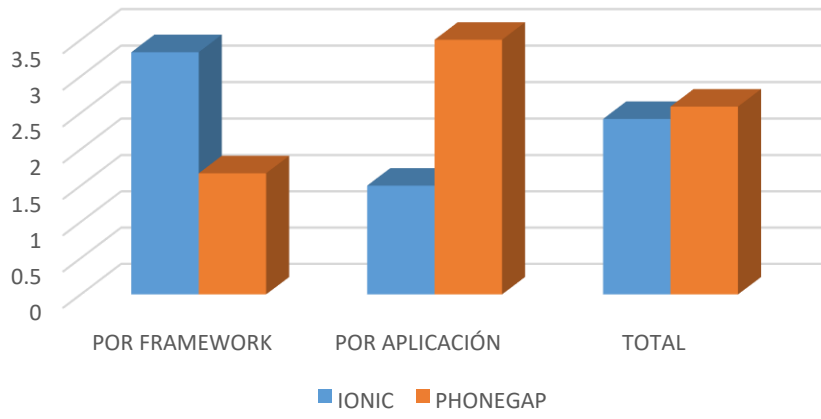
APLICACIONES MÓVILES HÍBRIDAS	PUNTAJE OBTENIDO (5 - GANADOR, 0 - PERDEDOR)			TOTAL	PROMEDIO
	CONSUMO DE MEMORIA RAM	% CPU	TIEMPO DE RESPUESTA		
IONIC	5	5	0	10	3.33
PHONEGAP	0	0	5	5	1.67

Resumen general de la comparativa

TECNOLOGÍA	VARIABLES		SUB-TOTAL	TOTAL	GANADOR EN PORCENTAJE
	FRAMEWORK	APLICACIÓN			
IONIC	3.33	1.50	4.83	2.415	48%
PHONEGAP	1.67	3.50	5.17	2.585	52%

Se muestra un gráfico con una visión global sobre la comparativa en mención, teniendo en cuenta la mínima diferencia 4% de superioridad y como posible mejor implementación con el framework Phonegap.

GRÁFICO RESUMEN FRAMEWORK CON MEJOR IMPLEMENTACIÓN



CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN

La propuesta de desarrollo parte de la premisa de que cada plataforma tiene su propio entorno de desarrollo, arquitectura, herramientas, el mercado de la distribución y el público objetivo, y cada enfoque de desarrollo tiene sus propias características inherentes a la forma en cómo es el diseño, desarrollo y distribución de aplicaciones.

Algo que se debe hacer por el equipo de desarrollo es decidir qué plataforma para desarrollar, siempre teniendo en cuenta que, al no crear la aplicación a una plataforma, puede ser ignorado en gran parte del mercado, lo que en última instancia, reducir el ámbito de aplicación y por lo tanto los beneficios de la aplicación.

Para el desarrollo de diversas aplicaciones para cada plataforma nativa, tiene que pasar por todo el proceso de desarrollo de nuevo porque para cada plataforma hay una arquitectura distinta que debe ser seguida, los estándares de codificación y la interfaz de usuario y las API propios. Por lo tanto, cada aplicación debe estar codificada considerando las diferentes arquitecturas, componentes y directrices, a la medida, tanto como sea posible para ser la misma aplicación en diferentes plataformas y finalmente generar un nuevo ejecutable y distribuirlo en las diversas tiendas de aplicaciones.

Por ejemplo, hay componentes que no existen en todas las arquitecturas

(navigationController y tabBarController presente en plataforma de IOS, por ejemplo) y se debe considerar en el diseño de la aplicación para otras plataformas.

Sin embargo, cuando se piensa en el desarrollo multiplataforma, el proyecto se convierte en agnóstico a la plataforma, ya que el proyecto se realiza pensando sólo en una arquitectura, un entorno de desarrollo y un único conjunto de herramientas, más no, de acuerdo a las características de cada plataforma. Después de desarrollar la aplicación principal, pueden realizarse personalizaciones para adaptar mejor la aplicación para cada distribución plataforma de destino, creando así un producto final verdadero.

A continuación se mostrará un resumen las principales ventajas y desventajas de cada tipo de desarrollo para una mejor comprensión de cada enfoque.

	VENTAJAS	DESVENTAJAS
NATIVO	<ul style="list-style-type: none"> · Explora todas las capacidades de los dispositivos. · Un rendimiento más alto. · Ofrece experiencia nativa la facilidad de uso para el usuario. · Estrecha integración con el sistema y otras aplicaciones. 	<ul style="list-style-type: none"> · Se necesita una aplicación por la plataforma. · Más caro y más difícil de mantener, por tener que mantener varias aplicaciones diferentes. · Se requiere de varios dominios entornos y lenguajes para cada plataforma.

<p>MULTIPLATAFORMA</p>	<ul style="list-style-type: none"> · Todo lo que se necesita para dominar un idioma y el medio ambiente. · Desarrolla solamente un código y se puede distribuir en varias tiendas de aplicaciones, que hace que el alcance de la aplicación es mayor. · Reduce el coste, tiempo y esfuerzo para desarrollar y mantener. 	<ul style="list-style-type: none"> · No tiene acceso a todas las funciones del dispositivo. · Menor rendimiento en comparación con el nativo, interpretado y compilado idiomas. · Carece de la misma usabilidad y experiencia de usuario de las aplicaciones nativas. · No tiene las últimas actualizaciones liberadas por el
-------------------------------	--	---

		<p>sistema operativo, ya que para cada actualización del sistema operativo, que necesita ser hecho una actualización de la plataforma.</p>
--	--	--

Tabla 12: Ventajas y desventajas de los enfoques de desarrollo nativos y multiplataforma

En este capítulo se describe la investigación llevada a cabo y los procedimientos realizados con el fin de alcanzar los objetivos del trabajo, los detalles de cómo fue el desarrollo de la misma y, por tanto, proporcionar insumos para su correcta reproducción o continuación en el futuro por otros investigadores.

Este trabajo se basa en una comparación entre dos frameworks para el desarrollo móvil, para responder a la pregunta de investigación problema se hará sobre las aplicaciones móviles y en los enfoques de desarrollo

de plataformas nativas. Para ello, se van a utilizar los artículos y materiales en línea para definir la forma, por lo tanto, se hace una comparación entre los dos modos de este desarrollo y después de estudiar y definir cómo es este software específico de desarrollo, se hará una plataforma de búsqueda para el desarrollo de plataforma cruzada.

También hizo la implementación de una aplicación utilizando una herramienta multiplataforma, en este caso, el PhoneGap. Se concluye que existe una necesidad de evaluar los requisitos de las aplicaciones que se creen para tomar la mejor decisión sobre qué enfoque elegir. Vale la pena señalar, sin embargo, que ambos frameworks son de desarrollo nativo para una comparación más precisa entre las dos formas de desarrollo abordado en este trabajo.

Se hace un estudio comparativo sobre la plataforma de desarrollo, sin embargo, no hace una comparación práctica entre los dos enfoques. La conclusión a la que se llega es que el desarrollo de híbridos es la solución más probable en el choque nativa vs multiplataforma, sin embargo, uno tiene que evaluar las necesidades de la aplicación y de negocios.

Se hizo un estudio de las posibles ventajas y desventajas de enfoque multiplataforma desde la perspectiva de tres diferentes Stakeholders: usuario, desarrollador y proveedor de la plataforma. La investigación llegó a la conclusión de que los usuarios son las principales guías del mercado y sus preferencias es que deben definir el futuro del desarrollo

móvil. Sin embargo, no se ha hecho ningún estudio práctico para comparar ambos enfoques.

Ejemplo de planificación de uso

En las siguientes subsecciones, se presentan los criterios utilizados por los autores para la selección de proyectos a ser recreada y la herramienta que se utilizará para el estudio del uso, así como una programación del desarrollo del proyecto.

La selección de proyectos

Para la realización del ejemplo de su uso, la aplicación Mini Farma fue elegido, creado de forma nativa para la plataforma iOS. La elección fue por factores enumerados a continuación:

- a) Es propiedad del autor de este trabajo, lo que facilita la obtención y comprensión del código.
- b) Explora las características nativas de los dispositivos como la ubicación, que serán explicados más adelante.
- c) Es una aplicación sencilla pequeña, lo que hace que sea ideal para hacerse de nuevo dentro del tiempo de ejecución normal.

La selección de plataformas

Actualmente, hay muchas herramientas y marcos para el desarrollo de aplicaciones móviles multiplataforma. Entre ellos, la elección de la base IONIC fue hecha por algunos factores, que se enumeran a continuación:

- a) Es un marco bajo una licencia de software libre.
- b) Cuenta con un entorno de soporte gratuito y robusto, con muchas herramientas y tutoriales disponibles.
- c) Utilice AngularJS y Córdoba son dos marcos estables, bien conocidos y utilizados y mantenidos por dos grandes empresas.
- d) Tiene muchos plugins disponibles y una comunidad muy activa para resolver problemas.
- e) Puede crear la apariencia y facilidad de uso al lado de las plataformas nativas.
- f) Tiene una arquitectura diseñada para optimizar el rendimiento de los dispositivos móviles.

La planificación del desarrollo

Después de definido el proyecto y la herramienta a utilizar, el siguiente paso es planificar la forma en que se llevará a cabo el desarrollo de la aplicación multiplataforma. El proyecto pasó por algunos pasos de replicación, y citó detalla a continuación:

- a) Configuración del marco IONIC y dependencias de las máquinas desarrolladoras.
- b) Definición e implementación del mismo entorno de desarrollo para ambos desenvolvedores.
- c) Familiarización con el marco IONIC y su entorno mediante la creación de proyectos de prueba y tutoriales.
- d) Familiarización con AngularJS y su arquitectura.
- e) Separación aplicación Mini Farma en sus principales características para ser recreada.
- f) Notificar las observaciones sobre las diferencias, dificultades, características, ventajas y desventajas.

Análisis exploratorio de la planificación

La puesta en práctica de comparación de características que se hizo de acuerdo a la siguiente lista:

- a) La selección de características comunes en aplicaciones de gran tamaño.
- b) Contextualización y explicación de la función a través de textos.
- c) Buscar funcionalidad posibilidad de desarrollo en IONIC.

- d) El desarrollo de código para la funcionalidad en IONIC;
- e) La minería o el desarrollo de código nativo para la misma funcionalidad.
- f) Evaluación de las características relativas a la posibilidad y la complejidad del desarrollo.
- g) Creación y aplicación de un cuestionario basado en los resultados de funcionamiento.
- h) Evaluar la comprensión de los profesionales sobre la situación actual de desarrollo móvil de plataforma cruzada.

Ejemplo de Uso

Con el fin de comparar el desarrollo de aplicaciones móviles utilizando el enfoque nativa multiplataforma, se llevó a cabo un estudio preliminar. Para ello, una aplicación hecha en iOS nativas se ha recreado utilizando el marco IONIC. A continuación, la aplicación seleccionada se detalla en términos de características y recursos utilizados.

Descripción del proyecto seleccionado

Mini Farma es una aplicación creada inicialmente para la plataforma iOS que sirve para controlar los medicamentos que las personas tienen en casa.

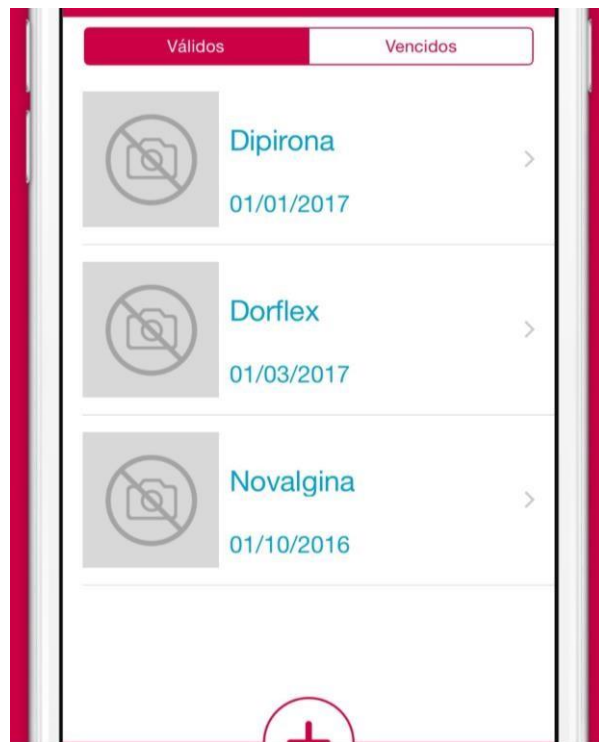


Figura 5: aplicación Home Mini Farma.

Fuente: Elaboración propia

La aplicación utiliza características nativas del sistema, que se describen a continuación.

· **Localización geográfica:** Establece la ubicación geográfica del dispositivo que el usuario está utilizando para guardar la ubicación farmacia donde compró un medicamento, si es necesario, para explicar a alguien en la farmacia. También es posible sobre la base de esta ubicación de la farmacia y al usuario trazar una ruta que le llevará directamente a la farmacia en cuestión.

· **Notificaciones locales:** A diferencia de las notificaciones Push, notificaciones locales son creadas y programadas en las notificaciones de dispositivos centrales, y el sistema se encarga de entregarlos adecuadamente de acuerdo con los parámetros definidos por la aplicación. En el caso del Mini Farma, las notificaciones se utilizan para recordar al usuario, la fecha y la hora correcta, que debe tomar sus medicamentos.

· **Enlace:** Las notificaciones pueden contener acciones que realizan un determinado bloque de código de la aplicación. En el Mini Farma, una de las posibles notificaciones está advirtiendo pequeña cantidad o recurso agotado en el caso de que el fármaco sea terminado. Cuando se envía esta notificación, se puede hacer un enlace directo por la acción de la notificación al número farmacia registrada en la aplicación. Por lo tanto, el usuario puede solicitar una nueva cantidad de medicamento directamente a la farmacia.

- **Cámara:** Para facilitar la identificación de los medicamentos, se puede tomar una foto con la cámara del dispositivo para cada medicamento registrado. Además de una imagen a la propia medicina, se puede tomar una foto de su ingreso, en su caso.

- **Rollo de la cámara:** Si ya tiene una imagen para ayudar a identificar el medicamento o la misma receta guardada en el rollo de la cámara, se puede elegir la foto sin tomar una nueva.

Excepto por la conexión, todos los demás recursos del sistema nativo debe seguir necesariamente el usuario que se les permita funcionar. Si el usuario no autoriza la aplicación se reduce la funcionalidad.

Como se utilizó la base de datos SQLite, a través del marco y de código abierto FMDB externa, disponible en Github.

La arquitectura del sistema se creó basado en el patrón de diseño MVC, que se muestra en la figura, que tiene una capa DAO (Data Access Object) para comunicarse con la base de datos. Sin embargo, el tradicional MVC difiere en algunos aspectos del MVC que se utiliza en iOS.

En iOS, la MVC detrás de un controlador conectado a la clase de vista, que se llama *ViewController*, responsable de crear el puente entre la interacción del usuario con la vista clases de modelos. Por lo tanto, los ViewControllers

tienen la responsabilidad de pasar la entrada del usuario para controlar el modelo de vista y para presentar los resultados.

A diferencia de MVC tradicional, donde el padre transmite sólo la información que está entrando en el límite de la aplicación, los ViewControllers tienen, además de esta responsabilidad también la responsabilidad para crear instancias y gestionar el punto de vista con respecto a la organización de los elementos y presentación de la información.

Otra observación acerca de las diferencias entre los dos los AT es que en MVC tradicionales puede ser un controlador sólo para cada modelo o un controlador de modelo, pero no conectado directamente a un punto de vista, sin embargo, hay una ViewController en IOS para vista y también puede ser uno de los padres en relación con cada modelo como el modelo tradicional de la MVC.

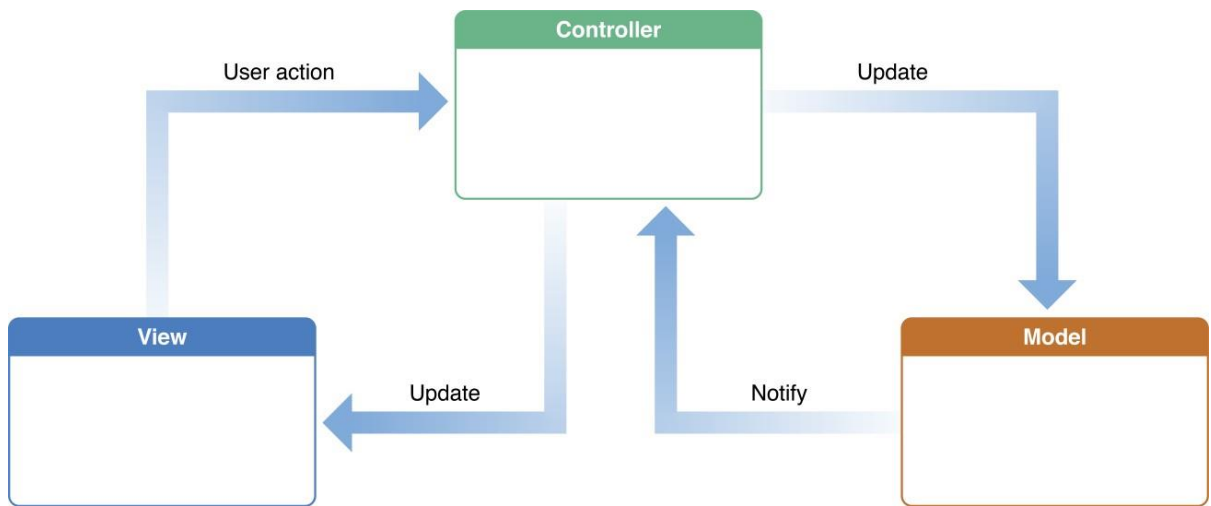
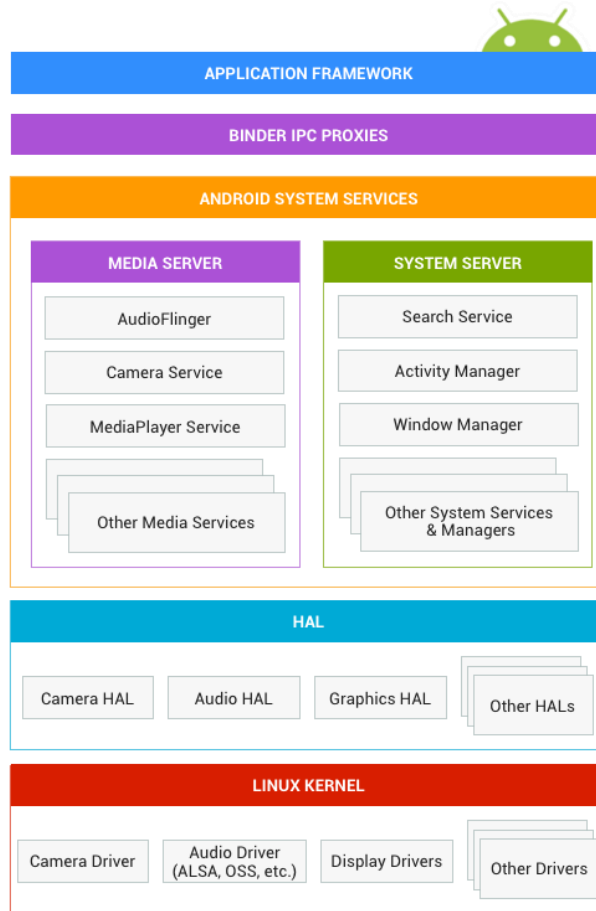


Figura 6: Standard Model-View-Controller.

Fuente: Elaboración propia.

Diagrama de arquitectura aplicada

La arquitectura se basa en capas mostrados en la figura.



CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- a) Se analizaron 10 frameworks existentes en el mercado seleccionándolos por las características que ofrecen y las plataformas de desarrollo utilizadas.
- b) Se seleccionaron 2 frameworks para el desarrollo de la

comparativa, la selección de estos frameworks se definió evaluando las mejores bondades de cada framework.

- c) Se implementó una aplicación basada en los frameworks seleccionados, la aplicación resultante se describe en la propuesta de desarrollo y muestra la aplicación del framework.
- d) Los resultados de la presente investigación se detallan en el capítulo correspondiente, los resultados muestran una favorable respuesta ante el framework IONIC.

6.2. Recomendaciones

Se recomienda a la empresa CODEX PERU desarrolladora de aplicaciones móviles que como trabajo a futuro se puede evaluar empíricamente herramientas para el desarrollo de la plataforma en tecnologías web, a través del desarrollo de aplicaciones y comparándolas con las aplicaciones desarrolladas de forma nativa.

BIBLIOGRAFÍA

- Aimacaña T, C. (2000). *Interfaz de usuario*. Recuperado el 22 de marzo de 2008, de <http://www.monografias.com/trabajos6/inus/inus.shtml>
- Appelquist, D. (2007). *Diseñar para la web móvil*. Recuperado el 8 de mayo de 2008, de <http://www.pixelovers.com/p/disenar-para-la-webmovil-16029>
- Barth, N. (2014). Análisis comparativo de herramientas de desarrollo para aplicaciones móviles. Universidad Regional de Blumenau.
Recuperado de: http://dsc.inf.furb.br/arquivos/tccs/monografias/2014_1_niksonbarth_monografia.pdf
- Bezerra, G. (2016). Desarrollo de aplicaciones multiplataforma usando PhoneGap. Recuperado de: <http://eumed.net/cursecon/ecolat/br/16/phonegap.html>
- Cevallos, E. (2014). Estudio de caso sobre Aplicaciones Móviles UX: Efecto del uso del marco de desarrollo multiplataforma. Tesis (Tesis de Maestría) - Universidad Politécnica de Madrid. Recuperado de: http://oa.upm.es/30422/1/EMSE-2014-05_Esteban_Angulo-1.pdf
- Jobe, W. (2013). Aplicaciones Nativas vs. Aplicaciones Móviles. *International Journal of Interactive Mobile Technologies*.

Recuperado de: [Http://online-journals.org/i-jim/article/view/3226](http://online-journals.org/i-jim/article/view/3226)

Maldonado, T. (1994). *Lo real y lo virtual*. Barcelona: Editorial Gedisa.

Meier, R. (2015). Crear mejores experiencias de usuario en Google Play. Recuperado de: <http://android-developers.blogspot.com.br/2015/03/crear-mejor-userexperiencias-on.html>

Nielsen, J. (2000). *Usabilidad: Diseño de sitios web*. Madrid: Editorial Prentice Hall.

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

Papajorgji, P. (2015). *Automatizado Sistemas de la empresa para el rendimiento y Maximización de negocios*. 1 edición. ed. Hershey, PA: IGI Global, 2015. ISBN 978-1-4666-8841-4.

Prezotto, E. (2014). Estudio de las diversas plataformas de desarrollo de aplicaciones móviles híbridas.

Rebouas, M. (2016). Un estudio empírico sobre el uso del lenguaje de programación Swift. v. 1, p. 634-638.

Rodríguez, A. & Baldrich, R. (2015). Diseño e Implementación de Una

Aplicación en un entorno multidispositivo de HTML5.

Shakshuki, E. (2013). Marco de componentes para crear aplicaciones móviles multiplataforma. Vol. 19, p. 1004 - 1011, enero 2013. ISSN 1877 a 0.509. Recuperado de: <http://www.sciencedirect.com/science/article/pii/S1877050913007485>

ANEXOS

Anexo 01: Cuestionario

- 1.- ¿Qué FRAMEWORKS de desarrollo móvil utilizo?
- 2.- ¿Es posible utilizar autenticación de usuario con Facebook utilizando tecnología multiplataforma?
- 3.- ¿Es posible realizar consumo de datos vía WEBSERVICE utilizando tecnología multiplataforma?
- 4.- ¿Es posible la utilización de lector biométrico para fines de autenticación de usuario utilizando tecnología multiplataforma?
- 5.- ¿Considerando la necesidad de extraer metadatos de archivos de medios es posible extraerlos utilizando tecnologías multiplataforma?
- 6.- ¿Es posible enviar e-mail y/o SMS directamente por la aplicación si éste fue desarrollado en multiplataforma?
- 7.- ¿Es posible utilizar tecnologías multiplataforma para la creación de Widgets?