



UNIVERSIDAD CENTRAL DEL ECUADOR

FACULTAD DE INGENIERÍA, CIENCIAS FÍSICAS Y MATEMÁTICA

CARRERA DE INGENIERÍA INFORMÁTICA

**ESTUDIO COMPARATIVO DE METODOLOGÍAS DE DESARROLLO ÁGIL EN BASE AL
DESARROLLO DE UNA APLICACIÓN MÓVIL, MODALIDAD PROYECTO DE
INVESTIGACIÓN, PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO INFORMÁTICO**

**TRABAJO DE GRADUACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO INFORMÁTICO**

AUTOR: WILMER ALBERTO ZAMBRANO GARCÍA

TUTOR: ING. LUIS FELIPE BORJA BORJA MSc.

QUITO, 14 DE FEBRERO

2017

AUTORIZACIÓN DE LA AUTORÍA INTELECTUAL

DERECHOS DE AUTOR

Yo, Wilmer Alberto Zambrano García en calidad de autor del proyecto de investigación estudio comparativo de metodologías de desarrollo ágil en base al desarrollo de una aplicación móvil, autorizo a la Universidad Central del Ecuador hacer uso de todos los contenidos que me pertenecen o parte de los que contiene esta obra, con fines estrictamente académicos o de investigación.

Los derechos que como autores me corresponden con excepción de la presente autorización, seguirán vigentes a mi favor, de conformidad con lo establecido en los artículos 5, 6, 8; 19 y demás pertinentes de la Ley de Propiedad Intelectual y su Reglamento.

Asimismo, autorizo a la Universidad Central del Ecuador para que realice la digitalización y publicación de este proyecto de investigación en el repositorio virtual, de conformidad a los dispuestos en el Art. 144 de la Ley Orgánica Educación Superior.

En la ciudad de Quito, a los 14 días del mes de Febrero del 2017



Wilmer Alberto Zambrano García

CC: 1311444812

Teléfono: 0968808831

Email: zagarc90will@gmail.com

CERTIFICACIÓN DEL TUTOR

APROBACIÓN DEL TUTOR DEL TRABAJO DE TITULACIÓN

Yo, Luis Felipe Borja, en calidad de tutor del trabajo de titulación investigación estudio comparativo de metodologías de desarrollo ágil en base al desarrollo de una aplicación móvil, elaborado por el estudiante Wilmer Alberto Zambrano García de la Carrera de Ingeniería Informática, Facultad de Ingeniería, Ciencias Físicas y Matemática de la Universidad Central del Ecuador, considero que el mismo reúne los requisitos y méritos necesarios en el campo metodológico y en el campo epistemológico, para ser sometido a la evaluación por parte del jurado examinador que se designe, por lo que APRUEBO, a fin de que el trabajo integrador sea habilitado para continuar con el proceso de titulación determinado por la Universidad Central del Ecuador.

El trabajo presentado pasó el control antiplagio URKUND

En la ciudad de Quito, a los 14 días del mes de Febrero del 2017



Ing. Luis Felipe Borja MSc

CC: 1711916088

Teléfono: 0998131629

Email: lborja@uce.edu.ec

INFORME DE LOS REVISORES



UNIVERSIDAD CENTRAL DEL ECUADOR
FACULTAD DE INGENIERÍA, CIENCIAS FÍSICAS Y MATEMÁTICA
DIRECCIÓN CARRERA DE INGENIERÍA INFORMÁTICA

Quito, 01 de febrero de 2017.
Oficio 025-2017-DC.

Señores:

Ing. Wagner Lucero

Ing. Mario Morales

DOCENTES REVISORES DE PROYECTO INTEGRADOR/INVESTIGACIÓN

Presente.-

Estimados Docentes:

Una vez que está culminado el proyecto INTEGRADOR titulado "**Estudio comparativo de Metodologías de Desarrollo ágil en base al desarrollo de una aplicación móvil**", elaborado por el (los) postulante (s) ZAMBRANO GARCÍA WILMER ALBERTO, me permito solicitar de manera comedida ***procedan a evaluar el proyecto*** señalado de acuerdo al formato remitido, en un plazo máximo de 10 días y entregar en la Dirección de la Carrera para continuar con el trámite respectivo.

Agradezco su atención.

Atentamente,

Ing. Boris Herrera Flores., MSc.
**DIRECTOR DE LA CARRERA DE INGENIERÍA
EN INFORMÁTICA.**



Anexo: Formato para evaluación.
Pacho Burbano.



UNIVERSIDAD CENTRAL DEL ECUADOR
FACULTAD DE INGENIERÍA, CIENCIAS FÍSICAS Y MATEMÁTICA
DIRECCIÓN CARRERA DE INGENIERÍA INFORMÁTICA

UNIDAD DE TITULACIÓN

RESULTADO DEL PROYECTO DE INVESTIGACIÓN/PROYECTO INTEGRADOR

CARRERA DE: INGENIERÍA INFORMÁTICA

FECHA: 09 de febrero de 2017.

POSTULANTE (S): ZAMBRANO GARCÍA WILMER ALBERTO

PROYECTO: "Estudio comparativo de Metodologías de Desarrollo ágil en base al desarrollo de una aplicación móvil".

CALIFICACIÓN:

TRIBUNAL	PROFESOR (A)	NOTA SOBRE CUARENTA	
		NUMEROS	LETRAS
REVISOR	Ing. Wagner Lucero	37,00	Treinta y Siete Coma Cero
REVISOR	Ing. Mario Morales	38,00	Treinta y Ocho Coma Cero
PROMEDIO		37,50	Treinta y Siete Coma Cincuenta
Ing. Boris Herrera MSc. DIRECTOR DE LA CARRERA			

Dr. Mario Guerra Burbano
SECRETARIO ABOGADO.



Elaborado por: PAUL BUSTAMANTE

DEDICATORIA

A:

Mi madre Alexandra por ser el motor de mi vida, por enseñarme en cada momento el significado de perseverancia y esfuerzo, por ser mi apoyo en cada paso que doy, por colmarme de infinito amor e inculcarme siempre valores y sacar lo mejor de mí, por ser mi maestra de la vida y mi héroe.

Por eso y mucho más este trabajo te lo dedico a ti, gracias por creer en mí, te amo mama.

AGRADECIMIENTO

Agradezco a Dios, por brindarme la vida y estar a mi lado en cada paso que doy, por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

Agradezco a mi familia mi papa Wilmer, mi mama Alexandra y mis hermanos Ray, Kira y Raysa por ser parte de mis esfuerzos, por apoyarme y quererme en cada momento, por ser mi respaldo y motivación.

A mis tías María Inés y Taty que han sido como una madre para mí toda mi vida y siempre han estado ahí apoyándome a mí y a mis hermanos.

A mis primos Gabriel y Andrés por ser como verdaderos hermanos, por compartir conmigo y brindarme su apoyo incondicional.

A mis amigos y compañeros de la universidad por compartir sus conocimientos y este camino profesional conmigo y estar siempre apoyándome.

A mi tutor y maestros que me han apoyado, brindado sus conocimientos y guiado en mi formación profesional a lo largo de esta etapa de mi vida.

Muchas gracias a todos ustedes, este trabajo fue sólo posible a todos ustedes.

CONTENIDO

AUTORIZACIÓN DE LA AUTORÍA INTELECTUAL.....	ii
CERTIFICACIÓN DEL TUTOR.....	iii
INFORME DE LOS REVISORES.....	iv
NOTAS.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO.....	vii
LISTA DE TABLAS.....	xi
LISTA DE FIGURAS.....	xii
RESUMEN.....	xii
ABSTRACT.....	xiii
INTRODUCCIÓN.....	1
1. MARCO TEÓRICO.....	3
1.1 Qué es metodología.....	3
1.2 Metodologías de desarrollo de software.....	3
1.3 Dispositivos móviles y metodología de desarrollo.....	8
1.3.1 Metodologías de desarrollo enfocadas a dispositivos móviles.....	8
2. METODOLOGÍA.....	13
2.1 Métodos teóricos.....	13
2.2 Métodos empíricos.....	13
2.3 Variables.....	13
2.4 Justificación del marco metodológico.....	14
3. EXPLORACIÓN DE DATOS SOBRE EL USO DE LAS METODOLOGÍAS ÁGILES.....	15
3.1 Exploración de datos en el marco internacional.....	15
3.2 Exploración de datos en el marco nacional.....	18
3.3 Elección de las metodologías a comparar.....	21
3.4 SCRUM.....	21
3.4.1 Teoría del control empírico.....	22
3.4.2 Sprint.....	23
3.4.2.1 Objetivo del sprint (Goal Sprint).....	23
3.4.2.2 Eventos formales.....	23
3.4.2.3 Artefactos de SCRUM.....	24
3.4.2.4 El equipo SCRUM (SCRUM Team).....	24

3.5	KAMBAN.....	25
3.5.1	Multitareas.....	26
3.5.2	Funcionamiento	27
3.6	XP (EXTREME PROGRAMMING)	27
3.6.1	Prácticas	28
3.6.2	Fases de XP.....	29
3.6.3	Roles.....	30
3.7	Mobile - D.....	31
3.7.1	Fases.....	33
3.8	Análisis de las metodologías	34
3.9	Implementación de la Metodología en la construcción del prototipo del aplicativo móvil .	40
4.	DISCUSIÓN	56
5.	CONCLUSIONES	58
6.	RECOMENDACIONES.....	60
7.	BIBLIOGRAFÍA.....	61
ANEXOS	65
ANEXO A	66
ANEXO B	68
ANEXO C	76
ANEXO D	78
ANEXO E	88
ANEXO F	91
ANEXO G	92
ANEXO H	96

LISTA DE TABLAS

<i>Tabla 1. Cuadro comparativo de las metodologías ágiles vs. las metodologías tradicionales.</i>	7
<i>Tabla 2. Características de Agilidad en el desarrollo de software para móviles.....</i>	10
<i>Tabla 3. Cuadro de comparaciones de metodologías ágiles.</i>	37
<i>Tabla 4. Tabla de ponderaciones de las metodologías ágiles vs características para el desarrollo de aplicaciones móviles.....</i>	38
<i>Tabla 5. Especificaciones del Dispositivo GPS</i>	49
<i>Tabla 6. Indicadores del dispositivo GPS.....</i>	50
<i>Tabla 7. Comando de funcionamiento del dispositivo GPS.....</i>	50
<i>Tabla 8. Requerimiento de Dispositivo Móvil.....</i>	51
<i>Tabla 9. Requerimiento de Servidores para API RESTful.</i>	52

LISTA DE FIGURAS

<i>Figura 1. Esquema general de la Metodología Ágil</i>	8
<i>Figura 2. Uso de S.O. en Dispositivos Móviles en el Ecuador</i>	9
<i>Figura 3. Las principales mejoras que se evidenciaron al implementar metodologías ágiles</i>	15
<i>Figura 4. Uso de las metodologías ágiles</i>	16
<i>Figura 5. Técnicas usadas de metodologías de desarrollo ágil</i>	16
<i>Figura 6. Causas que impiden adoptar las metodologías ágiles</i>	17
<i>Figura 7. Perspectiva de cómo miden el éxito de sus iniciativas ágiles</i>	17
<i>Figura 8. Uso de las metodologías ágiles (Datos locales)</i>	18
<i>Figura 9. Las principales mejoras que se evidenciaron al implementar metodologías ágiles (Datos Locales)</i>	19
<i>Figura 10. Causas que impiden adoptar las metodologías ágiles (Datos locales)</i>	19
<i>Figura 11. Perspectiva de cómo miden el éxito de sus iniciativas ágiles (Datos locales)</i>	20
<i>Figura 12. Técnicas usadas de metodologías de desarrollo ágil (Datos locales)</i>	20
<i>Figura 13. Esquema de trabajo SCRUM</i>	24
<i>Figura 14. Tablero KANBAN</i>	25
<i>Figura 15. Efecto del desarrollo de tareas paralelas</i>	26
<i>Figura 16. Esquema de las fases de XP</i>	31
<i>Figura 17. Modelo de Iteraciones en XP</i>	31
<i>Figura 18. Fases y etapas de Mobile-D</i>	33
<i>Figura 19. Desarrollo iterativo e incremental en la fase de producción</i>	34
<i>Figura 20. Fase de Estabilización</i>	34
<i>Figura 21. Arquitectura del sistema</i>	44
<i>Figura 22. Arquitectura del Servidor Web – API RESTful</i>	45
<i>Figura 23. Arquitectura de la Aplicación Móvil</i>	45
<i>Figura 24. Interface de ingreso a la aplicación</i>	46
<i>Figura 25. Interfaz de principal de la aplicación</i>	46
<i>Figura 26. Interfaces complementarias de la aplicación</i>	47
<i>Figura 27. Interfaces utilitarias</i>	47
<i>Figura 28. Componentes de la aplicación</i>	48
<i>Figura 29. Diagrama de secuencia</i>	48
<i>Figura 30. Resultado del test de carga y stress del servidor</i>	53
<i>Figura 31. Diagrama de comportamiento del servidor</i>	54

RESUMEN

ESTUDIO COMPARATIVO DE METODOLOGÍAS DE DESARROLLO ÁGIL EN BASE AL DESARROLLO DE UNA APLICACIÓN MÓVIL, MODALIDAD PROYECTO DE INVESTIGACIÓN.

Autor: Wilmer Alberto Zambrano García

Tutor: Luis Felipe Borja Borja

Este trabajo aborda y explora la eficiencia y eficacia del uso de las metodologías ágiles en la construcción de aplicaciones móviles mediante la comparación del uso aplicativo de las fases, tareas, principios y enfoque de las metodologías seleccionadas en este trabajo; y las características y limitaciones que conlleva en el proceso de desarrollo de software para dispositivos móviles. Para la realización este trabajo de investigación se aplicó a modo de ejemplo demostrativo, la metodología con mayor idoneidad MOBILE-D para el desarrollo de un prototipo de aplicación móvil.

PALABRAS CLAVES: METODOLOGÍAS ÁGILES / APLICACIONES MÓVILES / DESARROLLO DE SOFTWARE / INGENIERÍA DE SOFTWARE / PROCESO DE DESARROLLO / PRÁCTICAS Y MÉTODOS ÁGILES / DESARROLLO DE SOFTWARE PARA DISPOSITIVOS MÓVILES.

ABSTRACT

COMPARATIVE STUDY OF AGIL DEVELOPMENT METHODOLOGIES BASED ON THE DEVELOPMENT OF A MOBILE APPLICATION, RESEARCH PROJECT MODE

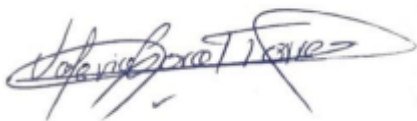
Autor: Wilmer Alberto Zambrano García

Tutor: Luis Felipe Borja Borja

This dissertation addresses and explores the efficiency and effectiveness of the use of agile methodologies on the development of mobile applications by comparing the application use of phases, tasks, principles and approach of the selected methodologies for the present work; and the characteristics and limitations that the software development process for mobile devices involves. To carry out the present research project the methodology with greater suitability MOBILE-D was applied as a demonstrative example for the development of a mobile application prototype

KEYWORDS: AGILE METHODOLOGIES / MOBILE APPLICATIONS / SOFTWARE DEVELOPMENT / SOFTWARE ENGINEERING / DEVELOPMENT PROCESS / AGILE PRACTICES AND METHODS / SOFTWARE DEVELOPMENT FOR MOBILE DEVICES.

I certify that the above and foregoing is a true and correct translation of the original document in spanish.



Valeria Coral Torres

IELTS Certified: (SENESCYT) 15CO009598CORV001A

IC: 1723207237

INTRODUCCIÓN

En los albores del siglo XXI la tecnología ha cambiado nuestro mundo, tanto que hoy los dispositivos móviles impregnan la vida diaria, dando un acceso incomparable a la comunicación y la información (Costa Sánchez, Díaz-González, y Videla Rodríguez, 2012). Como lo afirma la revista Forbes México (2015), tomado de Ericsson (2015) la cantidad de dispositivos móviles (7.300 millones) superan a la cantidad de la población mundial.

Hoy en día existen muchas guías de desarrollo ágil de software, las mismas no están centradas en el desarrollo de aplicaciones móviles ya que estas difieren del desarrollo de software clásico (Sifuentes, Tarelo, Saldaña, Demoss, & Madrazo, 2015). Según los investigadores (Blanco, Camarero, Fumero, Warterski, & Rodríguez, 2009), para el desarrollo de software se debe aplicar metodologías de desarrollo óptimas e idóneas para que dicho desarrollo tenga éxito, de una forma interesante, directa y eficaz; en este proyecto se realizará un análisis sobre las metodologías ágiles para determinar la idoneidad y pertinencia en la implementación de las misma para el desarrollo de aplicativos móviles.

Formulación del problema

¿Qué metodología de desarrollo ágil se puede implementar de forma exitosa en la elaboración de aplicaciones móviles?

Pregunta directrices

Las preguntas directrices que sean planteados para realizar este proyecto integrador son los siguientes:

- ¿Qué metodología o etapas de las mismas se acopla de mejor manera a la elaboración de aplicaciones móviles?
- ¿Hasta qué punto será eficiente la utilización de la metodología de desarrollo ágil en la elaboración de aplicaciones móviles?

Justificación

Hoy en día existen pocos estudios¹ sobre la utilización de la metodología de desarrollo ágil de software para la elaboración de aplicaciones móviles, por lo que es pertinente realizar este proyecto para generar una visión clara sobre la utilización y rentabilidad de metodologías ágiles para la elaboración de aplicativos móviles.

Objetivo general

Realizar un estudio comparativo de las metodologías ágiles para determinar qué metodología o etapas de la misma se acopla de mejor manera para el desarrollo óptimo de aplicaciones móviles o Apps.

Objetivos específicos

- Analizar las metodologías de desarrollo óptimo para la elaboración de aplicativos móviles o Apps.
- Implementación de la metodología de desarrollo ágil para la elaboración de aplicativos móviles.
- Crear una aplicación móvil para la búsqueda y localización de mascotas perdidas.

Alcance

- Realizar una tabla comparativa para el análisis de las metodologías ágiles.
- Implementar metodologías ágiles en el desarrollo de la aplicación móvil.

¹ Entre los estudios que se han realizado se encuentran: Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles (Amaya Balaguera, 2013), An Investigation into Mobile Application Development Processes: Challenges and Best Practices (Flora, Wang, & Chande, 2014), A Review and Analysis on Mobile Application Development Processes using Agile Methodologies (Flora & Chande, 2013), (P. Abrahamsson & VVT, 2010) y Agile Development Methods for Mobile Applications (Spataru, 2010)

MARCO TEÓRICO

1.1 Qué es metodología

La metodología es el convenio del método, la descripción, el análisis y la valoración crítica de los métodos de investigación. Ponce (1996), dice que “en su acepción más estricta, llámese metodología, al estudio sistemático de los métodos utilizados por una ciencia en su investigación de la realidad” (p.61).

Por otra parte Cortés e Iglesias (2004), sostienen que “la metodología es conjunto de procedimientos que nos enseña a dirigir determinado proceso de manera eficiente y eficaz para alcanzar los resultados deseados y tiene como objetivo darnos la estrategia a seguir en el proceso” (p.8).

Por lo que la metodología es el plan que nos permites conseguir los objetivos en el marco de una ciencia entendiéndose este como un conjunto de procedimientos que determinan una investigación, estudio o enmarcan una explosión doctrinal.

1.2 Metodologías de desarrollo de software

Hoy en día la competencia de las organizaciones demandan adaptarse rápidamente a los cambios, lo cual implica responder velozmente a estos cambios que surgen de las necesidades del día a día, y que en la industria del desarrollo de software existe un crecimiento vertiginoso de la utilización de nuevas tecnologías, procesos, modelos, etc.; en las que se acople a los objetivos de la misma, la “entrega de un *producto exitoso*” al cliente (Mariño, Godoy, y Alfonzo, 2014).

Existen guías de desarrollo de software avaladas que buscan la eficacia y éxito en la construcción del producto denominadas “*metodologías ágiles*”. La importancia de las metodologías ágiles es promover generalmente, un proceso de gestión de proyectos que fomenta el trabajo en equipo, la organización y compromiso propio, un conglomerado de mejores prácticas de ingeniería de software que posibilitan una entrega rauda de software de alta calidad, y un encauzamiento de negocio que alinea la construcción de software con las necesidades del cliente y los objetivos de la organización, ofreciendo una dirección a seguir

(Mariño et al., 2014); pero las cuales, cuando fueron concebidas no estaban centradas al desarrollo de aplicativos móviles.

Algunas definiciones encontradas designan a las metodologías ágiles de estas maneras:

Raya, dice que “La metodología ágil es una serie de métodos, procedimientos y técnicas para la gestión de proyectos de software que brotan en contraposición de las técnicas y procedimientos clásicos de gestión” (Raya, 2014).

Mientras que Vera define a la metodología ágil como “un marco metodológico de trabajo que propone posibilitar el incremento de la eficiencia en la producción y la calidad de software, tener la competencia de reacción al cambio en los productos y sus definiciones, y ofrecer la mayor complacencia posible al cliente, a través de la concesión anticipada y oportuna de software y la retroalimentación continua durante el desarrollo del producto” (Vera, 2014).

Poole define el desarrollo ágil “como aquel que, en semejanza con el desarrollo tradicional, genera ganancias de mayor flexibilidad, Retorno de Inversión más alto, realización más rápida del Retorno de Inversión, más alta calidad, mayor claridad, y paz sostenible” (Poole, 2009).

Y además Calderón, Dámaris, y Carlos, (2007) dicen que las metodologías ágiles nacen como respuesta para completar ese hueco metodológico, en que el sistema del entorno de desarrollo planteado que exigen los proyectos actuales es muy cambiante; dado estas dificultades para utilizar metodologías tradicionales con estas limitaciones de tiempo y flexibilidad, muchos equipos de desarrollo renuncian a la utilización de las buenas prácticas de la Ingeniería del Software. (p.3); esto en respuesta de las necesidades que surgen.

Por otra parte Navarro C., Fernández M., y Morales V., (2013) afirman que “las metodologías ágiles son flexibles, pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto, subdividiendo en proyectos más pequeños, en el que cada proyecto es tratado de manera independiente y desarrolla un subconjunto de características durante un periodo de tiempo corto. La comunicación con el cliente es constante. Los proyectos son altamente colaborativos y se adaptan mejor a los cambios.”(p. 31)

En general todos los autores citados hacen referencia a la necesidad de adaptarse a los nuevos estándares de requerimientos para la gestión de proyectos como la flexibilidad, comunicación constante entre el grupo de desarrollo y el cliente, retroalimentación, trabajo en equipo, cambios constante en los requerimientos de software y alta calidad del software entre otros; para poder sobrevivir al entorno tan cambiante y rápido que se vive en la industria de desarrollo de software.

El desarrollo de software mediante una metodología ágil es incremental, cooperativo, sencillo y adaptado; la cual debe satisfacer las características y necesidades de las aplicaciones móviles.

La importancia de las metodologías ágiles es promover generalmente un proceso de gestión de proyectos que fomenta el trabajo en equipo, la organización y compromiso propio, un conglomerado de mejores prácticas de ingeniería de software que posibilitan una entrega rápida de software de alta calidad, y un encauzamiento del negocio que enfila la construcción de software con los requerimientos del cliente y los ideales de la compañía.

Pero en definitiva toda metodología ágil se respalda o deben fundamentar en los **cuatro valores** señalados en el manifiesto ágil (Agile Alliance, 2001), el cual fue diseñado en respuesta a las nuevas exigencias y marcan diferencia con los sistemas tradicionales:

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

Los individuos y su interacción, por encima de los procesos y las herramientas.

Implica centrarse en la creación del equipo sobre el entorno, es más fácil establecer el equipo de trabajo y que éste estructure su entorno en base a las necesidades, que crear el entorno y aguardar que el equipo se adecue este. Además, las interacciones del equipo con el usuario final deben ser fluidas (miembro más del equipo), con un objetivo común, que es conseguir que el proyecto funcione y sea útil para él (Fernández González, 2013).

El software que funciona, frente a la documentación exhaustiva.

Este principio se basa en la “premisa que los documentos no pueden reemplazar ni generar el valor agregado que se otorga con la comunicación directa entre los individuos a través de la interacción con los prototipos” (Navarro C. et al., 2013). Por lo que lo mejor es intentar reducir la creación de documentos "inútiles", ya que estos no son mantenidos y peor consultados, sólo documentación que realmente sea útil, y que apoye de manera rápida la toma de decisiones en el desarrollo del software.

La colaboración con el cliente, por encima de la negociación contractual.

Señala la cooperación y la relación continua entre el cliente y el equipo de desarrollo, para que en la creación de los contratos por requerimiento según los entregables, eliminar los contratos donde supuestamente está “todos los requerimientos del cliente” ya que esto en la vida real esto no se da.

La respuesta al cambio, por encima del seguimiento de un plan.

Las organizaciones cambian constantemente, por lo que también cambia su flujo de trabajo para adaptarse a las nuevas necesidades, por lo que el desarrollo de software debe adaptarse a estos cambios para que el software se ajuste al modelo de negocio de las organizaciones. Por lo que esta habilidad debe ser un factor inherente al desarrollo de software el cual determina de gran manera el éxito o fracaso del proyecto.

Con las metodologías ágiles se pretende disminuir el impacto del trabajo que no es totalmente indispensable y aumentar la eficiencia de los individuos involucrados en el proyecto y como resultado de ello, disminuir el coste. Básicamente enfocándose en el resultado y en el valor que este aporta.

Los cuatro valores antes señalados se enmarcan en los 12 principios de la agilidad (Agile Alliance, 2001):

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo.
- Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Dada algunas de las características de desarrollo de software se muestra una tabla comparativa entre las metodologías tradicionales y metodologías ágiles (Tabla 1):

Tabla 1. Cuadro comparativo de las metodologías ágiles vs. las metodologías tradicionales. (Blanco, Camarero, Fumero, Warterski, y Rodríguez, 2009)

Area	Metodologías tradicionales	Metodologías ágiles
Desarrolladores	Predictivos Orientados a procesos	Adaptativos Orientados a personas
Proceso	Proceso rígido	Proceso flexible
Arquitectura	Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Comunicación	Poca comunicación con el cliente	Comunicación constante con el cliente
Requerimientos	Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación	Documentación extensa	Poca documentación

En la siguiente figura se muestra un esquema general de cómo trabajan las metodologías ágiles:

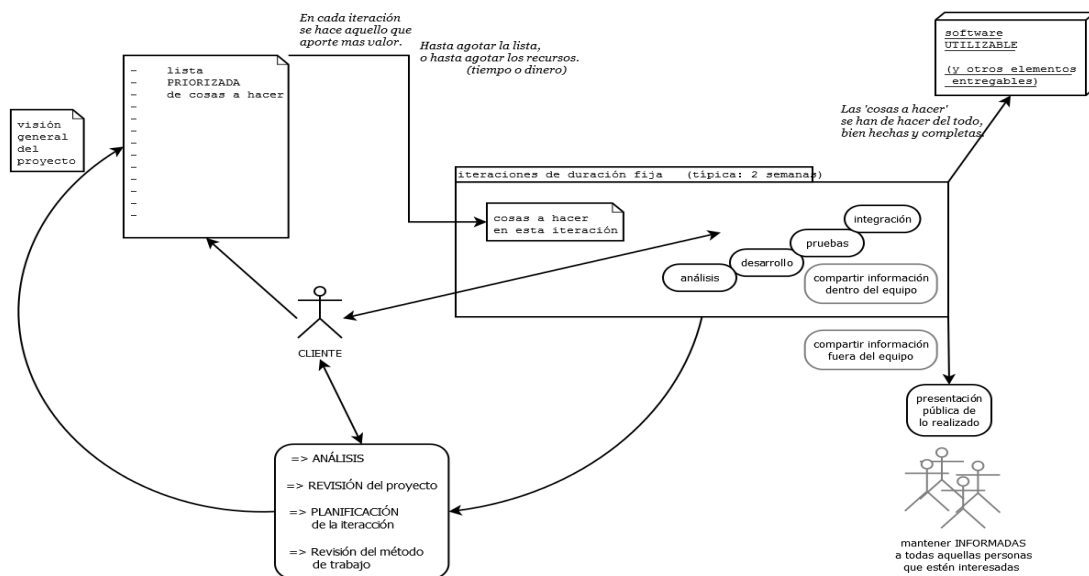


Figura 1. Esquema general de la Metodología Ágil.(Wikimedia Organization, 2012)

1.3 Dispositivos móviles y metodología de desarrollo

Los dispositivos móviles se presentan como “el nuevo motor de la sociedad de la información” y en el que la definición de estos es cada vez más difusa debido a la acelerada evolución que han tenido en las prestaciones de hardware-software y la accesibilidad de datos que resulta de la interrelación entre la Web2.0 y la “inteligencia colectiva”, configurando el mercado y generando un nuevo modelo de negocio. (Cerezo, 2010)

1.3.1 Metodologías de desarrollo enfocadas a dispositivos móviles

Las tecnologías móviles a medida que aumentan su capacidad, funcionalidad y potencias también se incrementa la capacidad de acceso hacia la información, por lo que la utilización de las tecnologías móviles potencia el uso de la información para mejorar la calidad de vida, ya sea en el aprendizaje, ayuda social, comercial, cultural, toma de decisiones, etc.

Según los datos, hasta mayo del 2013, un total de 1 261 944 ecuatorianos tenían Smartphone (Revista Lideres, n.d.); por lo que genera un mercado competitivo para el desarrollo de aplicaciones móviles o apps, y en el que 65.71% entre enero del 2015 y enero del 2016 usaban S.O. Android (figura 2), un porcentaje muy similar a escala mundial el 64.28% según

StatCounter Global Stats, (2015). El uso promedio de una persona en un móvil para descargar aplicaciones es de un 69% en Latinoamérica en 2013 (StatCounter Global Stats, 2015) y de las cuales lideran en descargas las categorías de juego, estilo de vida y productividad con un 63% entre los tres a nivel mundial mientras que en el entorno nacional los desarrolladores de app elaboran aplicativos para generar servicios en la industria ecuatoriana (Revista Lideres, n.d.).

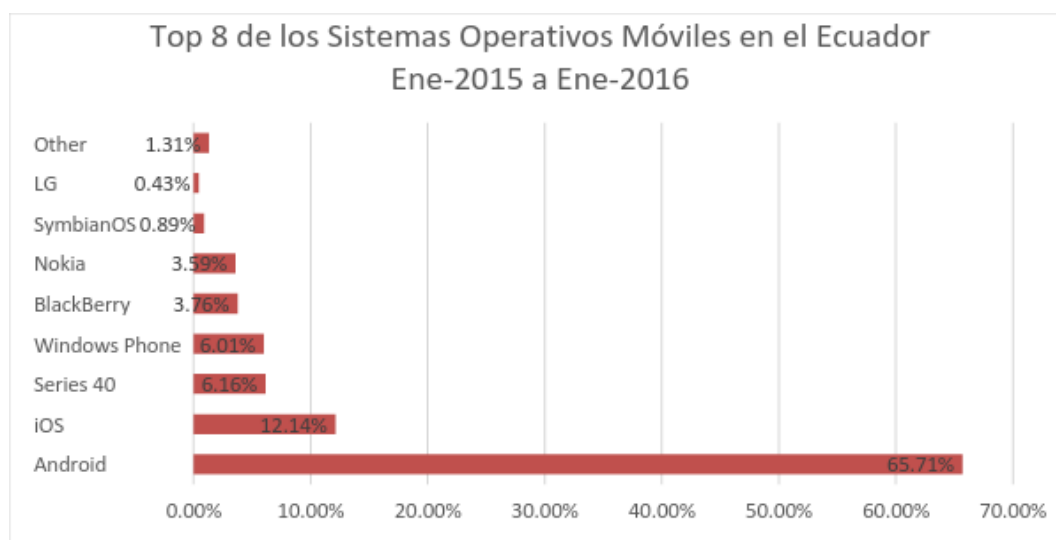


Figura 2. Uso de S.O. en Dispositivos Móviles en el Ecuador, tomado de (StatCounter Global Stats, 2015)

El mercado de las aplicaciones en los últimos se expande rápidamente día con día, existe una amplia gama de variedad en las aplicaciones y de dispositivos móviles; las plataformas mejoran el rendimiento y la utilización de los recursos, convirtiéndose en una alternativa para muchos usuarios como herramienta de trabajo. Las características y restricciones para el desarrollo de aplicaciones móviles es diferente en cada plataforma y estas a su vez difieren del desarrollo tradicional.

Las características distintivas para el desarrollo de aplicaciones móviles según (Abrahamsson, P., Hanhineva, Hulkko, Ihme, y Korkala, 2004) son: un alto nivel de competitividad; el tiempo de entrega necesariamente corta; y se añade la dificultad para identificar actores y sus requisitos. (Spataru, 2010), en la siguiente tabla muestra las características de agilidad en el desarrollo de aplicaciones móviles enunciadas por Pekka Abrahamsson.

Tabla 2. Características de Agilidad en el desarrollo de software para móviles. (Pekka Abrahamsson, 2005)

Ideales de agilidad Característica	Razón fundamental	Software para móviles
Alta volatilidad en el entorno	Debido al gran cambio de requisitos, menos necesidades de un diseño inicial y planificación, se necesita para el desarrollo incremental e iterativo.	Alta incertidumbre, medio ambiente dinámico: Cientos de nuevo teléfonos móviles publican cada año.
Los equipos pequeños de desarrollo	Los equipos pequeños son capaces de reaccionar con mayor rapidez, compartir información, menor es la necesidad de documentación.	La mayoría del software móvil es desarrollado en micro o SME empresas (small and medium enterprises) y sus equipos de desarrollo también.
Identificación del cliente	Para evitar malos entendidos negocio.	Número potencialmente ilimitado de los usuarios finales. Cliente de negocios más fáciles de identificar.
Entorno de Desarrollo Orientado a objetos	La mayoría de las herramientas que apoyan el desarrollo ágil necesitan de plataformas de desarrollo orientados a objetos.	Por ejemplo, Java y C ++ utilizan.
Nivel de aplicación del software	Sistemas grandes requieren amplios mecanismos de comunicación y de verificación.	Los sistemas móviles son complejos y dependen en gran medida, las aplicaciones móviles pueden ser aplicaciones independientes
Los sistemas pequeños	Menos necesarios etapas de diseño y planificación	Tamaño de las aplicaciones móviles varían, pero en general son menos de 10000 líneas de código.
Ciclos de desarrollo cortos	Efectos de retroalimentación rápida	Ciclos de desarrollo varían. En general, las aplicaciones y los servicios móviles se pueden desarrollar dentro de 1-6 meses plazo

Mientras que como restricciones (Hayes, 2003) menciona a 2 tipos de limitaciones; la primera descritas como limitaciones inherentes y de evoluciones, haciendo referencia a la capacidad de memoria, ancho de pantalla, reducción en la capacidad de entrada de datos, reserva de energía, capacidad de procesamiento, y la segunda a las limitaciones de seguridad y ancho de banda (cobertura); en la cual el desarrollo de debe tratar de minimizar el impacto de estas restricciones; mientras que (P. Abrahamsson, 2005), dice que si las aplicaciones son pequeñas, las limitaciones de seguridad no son crítica y no es necesario satisfacer las limitaciones de interoperabilidad y fiabilidad.

Hoy en día existen muchas guías de desarrollo ágil de software, las mismas no están centradas en el desarrollo de aplicaciones móviles como SCRUM, XP, etc.; y que difieren del desarrollo de software clásico pero que son utilizadas ampliamente. Según los investigadores, para el desarrollo de software se debe aplicar metodologías de desarrollo óptimas e idóneas para que dicho desarrollo tenga éxito, de una forma interesante, directa y eficaz. Entre las metodologías enfocadas al desarrollo de aplicaciones móviles destaca Mobile-D, otro es Hybrid Methodology Design, este se ampara en una unión del desarrollo adaptativo de software (ASD) y el Diseño de Nuevos productos (NPD por sus siglas en inglés), el cual parte del ciclo de vida tradicional e incluyen una fase de comercialización; existe una propuesta de metodologías diseñadas para aplicaciones móviles y que se encuentra en etapa experimental denominada Mobile Development Process Spiral, el cual es un modelo impulsado por la usabilidad y toma como base el modelo espiral. (Amaya Balaguera, 2013)

(Flora, Wang, & Chande, 2014) define el desarrollo de aplicaciones móviles “Mobile Application Development (MAD)” como un desarrollo para dispositivos de bajo consumo, en los que se debe considerar factores tecnológico en él se manifiesta 3 características de este: hardware, software, y comunicaciones; también factores de usabilidad como tecnologías de entradas y diseño de iteración de usuario estos 2 factores se generan dado a la opulencia de dispositivos móviles que existen hoy en día, la incertidumbre de los requerimientos y las características de las distintas plataformas móviles.

(Flora et al., 2014) propone los siguientes puntos como temas críticos y que se deben considerar en el desarrollo de aplicaciones móviles:

- **Compatibilidad con las plataformas**

Se debe generar un gran esfuerzo y costo dado a la complejidad de las plataformas móviles, ya que se propina la fragmentación de hardware, software y herramientas tecnológicas necesarias para la construcción de una App.

- **Variación compleja y hardware**

Dado el exceso de modelos de móviles se generan distintas características de hardware para cada uno de estos como, procesamiento gráfico, capacidad de procesamiento, memoria, disponibilidad de batería, capacidad de memoria, etc. que son limitadas en comparación de las PCs. Esto hace que resulte difícil realizar una aplicación que trabaje sin errores según la autora (Flora et al., 2014).

- **Recursos sin experiencia**

Según Flora determinó que menos del 10% del equipo de desarrollo muestran conocimiento y experiencia en los recursos móviles que se necesitan para el desarrollo de las Apps.

- **Presupuesto y calendario**

Planteada la necesidad de ganar el mercado los desarrolladores trabajan con limitado tiempo y comúnmente con poco presupuesto lo que recae en un desarrollo de baja calidad.

- **Experiencia de usuario UX**

La iteración del usuario en una aplicación tiene mayor impacto en un dispositivo móvil.

- **Interfaz de Usuario (UI)**

Dadas las características de la pantalla de un dispositivo móvil se deben efectivizar el uso de los métodos de entrada y salida de datos como el keyboard, formularios, para ofrecer simplicidad y claridad de la información que se está utilizando.

- **Acceso a datos**

La accesibilidad a datos su obtención, sincronización y mantenimiento de estos sin conexión.

- **Complejidad de Pruebas**

La variedad de dispositivos móviles promueven que se realice no solo pruebas en emuladores sino también en los dispositivos como tal para determinar el éxito o fracaso de la aplicación.

- **Análisis de usuarios objetivos**

Poder identificar y analizar los usuarios objetivos y sus preferencias debido a que la competencia por ganar el mercado de las aplicaciones es muy grande.

METODOLOGÍA

Para la realización de esta investigación se utilizaron los siguientes métodos los cuales serán parte del marco de trabajo a emplear para la realización de este trabajo:

2.1 Métodos teóricos

Análisis y síntesis: Para caracterizar las metodologías ágiles en el desarrollo de aplicaciones móviles, en el diagnóstico del proceso de construcción de las aplicaciones móviles en la realidad y la valoración de los resultados de la investigación.

Inducción–deducción: En la búsqueda de la solución al problema y en la elaboración de los aportes de la investigación.

Histórico-lógico: Para valorar el desarrollo histórico del proceso de evolución de las metodologías ágiles y su aplicación.

2.2 Métodos empíricos

Encuesta: A programadores de software para determinar tendencias de uso metodologías en el trabajo.

Entrevista: A profesionales con experiencia en el desarrollo de software, para caracterizar el manejo e implementación de los procesos ágiles.

2.3 Variables

Variable independiente: Requerimientos o características deseables en la construcción de un aplicativo móvil.

Definición de la variable

Son las singularidades que se dan en la construcción de aplicativos móviles y que transforman el resultado en base a la metodología utilizada.

Variable Dependiente: Qué metodología ágil tiene mayor eficacia en la construcción de App.

Definición de variable

Es la manipulación sistemática de las variables independiente para determinar de qué modo incide en la variable dependiente, es decir saber el impacto / eficacia que produce en el desarrollo de aplicaciones móviles (MAD).

2.4 Justificación del marco metodológico

Se escogió este marco metodológico debido a que es un proyecto de evaluación (caso de estudio) en la que se investiga la aptitud de una metodología frente a las peculiaridades que surgen en la construcción de un aplicativo móvil, entre las limitaciones en la investigación se encuentra la falta de experiencia en el ámbito de la gestión de proyectos y desarrollo de aplicaciones móviles, por lo que las conjeturas realizadas se generaran en base a lo que declaran los pensadores e investigadores y profesionales con experiencia en ámbito de la gestión y desarrollo de aplicaciones móviles.

Se procederá a escoger un marco de trabajo o metodología ágil que se acople de mejor manera al desarrollo de aplicativos móviles en función de sus características, con la cual se ejecutará un prototipo de una aplicación móvil para evidenciar los resultados inferidos en el análisis.

EXPLORACIÓN DE DATOS SOBRE EL USO DE LAS METODOLOGÍAS ÁGILES

3.1 Exploración de datos en el marco internacional

Segun Garner, 2015 citado en (Sifuentes, Tarelo, Saldaña, Demoss, & Madrazo, 2016) predice que para el 2017 la demanda de desarrollo de software aumentara 5 veces más que la capacidad de desarrollo en la área de TI.

En referencia al marco internacional según la empresa *Versionone Agile Made Easier* (Version One, 2015) empresa dedicada al desarrollo de herramientas para la gestión ágil de proyectos, en la décima edición de la encuesta a nivel internacional realizada en el 2015 sobre el seguimiento de este tipo de metodologías en las empresas, indican que el 95% de las organizaciones de software abrazan las metodologías ágiles, de los cuales las principales mejoras que se evidenciaron al implementar metodologías ágiles fueron del 87% la capacidad para gestionar los cambio prioritarios, el 85% dijeron incrementar la productividad y el 84% acelerar la entrega del producto como mayores beneficios, reafirmando que el uso de estas metodologías gestionan los proyecto complejos de mejor manera que los tradicionales.

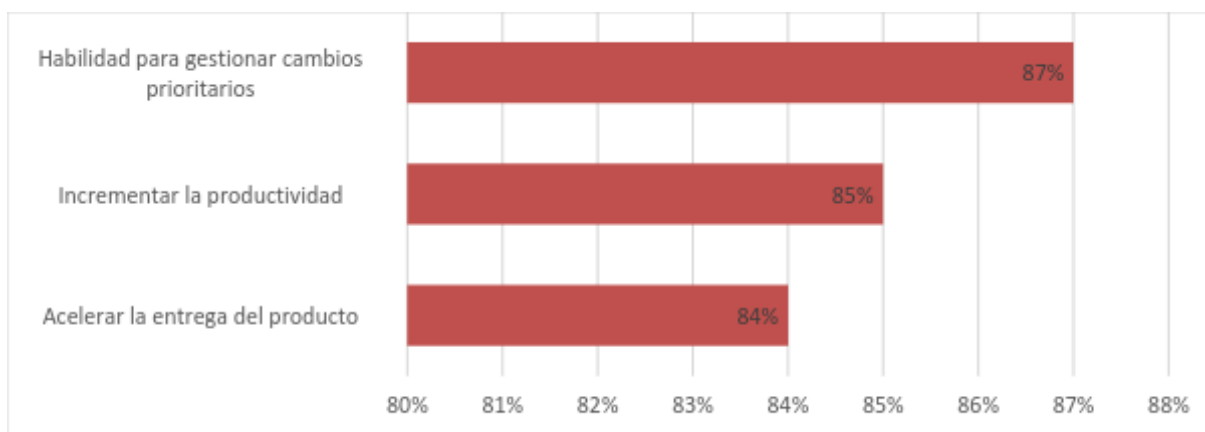


Figura 3. Las principales mejoras que se evidenciaron al implementar metodologías ágiles. (Version One, 2015)

Según esta misma encuesta manifiesta que la metodologías más usadas son SCRUM con el 58%, SCRUM/XP Hybrid 10% y Metodologías Híbridas (múltiples metodologías) el 8%, como las 3 metodologías de mayor uso, como vemos, de las personas encuestadas se entrevé que entre las metodologías existe una preferencia de uso de SCRUM, dando una brecha bastante grande con respecto a otras metodologías.

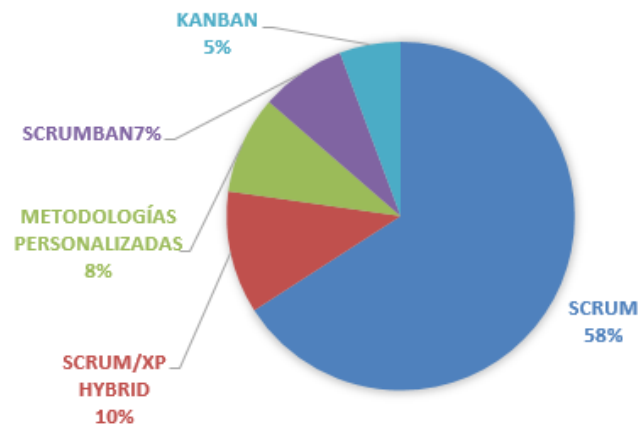


Figura 4. Uso de las metodologías ágiles. (Version One, 2015)

En esta misma encuesta señala que las técnicas más usada para el desarrollo del proyecto fueron las reuniones diarias con el 83%, la priorización de backlogs con el 82%, iteraciones cortas con el 79%, planificación de las iteraciones con 69% y retrospectivas con el 74% como se puede reflejar son técnicas de relevancia que están relacionada con el uso de la metodología SCRUM reafirmando la tendencia de utilización de la misma.

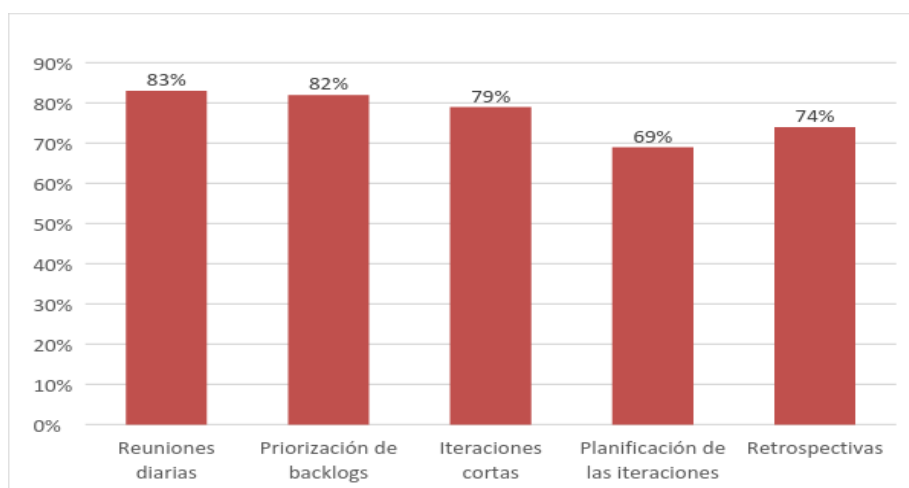


Figura 5. Técnicas usadas de metodologías de desarrollo ágil. (Version One, 2015)

La encuesta también indica que el 41% de las personas encuestadas señala que la causas que impide adoptar las metodologías ágiles es la falta de experiencias sobre uso de estas metodologías, mientras que el 46% dijeron que no poseer una cultura empresarial en base a los valores ágiles es la causa que impide adoptar estas metodologías, otro 38% manifestaron que la falta de apoyo de gestión es otra de las causa más importantes, como vemos la falta de experiencia sobre el uso de estas metodologías y la resistencia al cambio son los problemas que más influye con respecto a apropiarse de una metodología ágil.

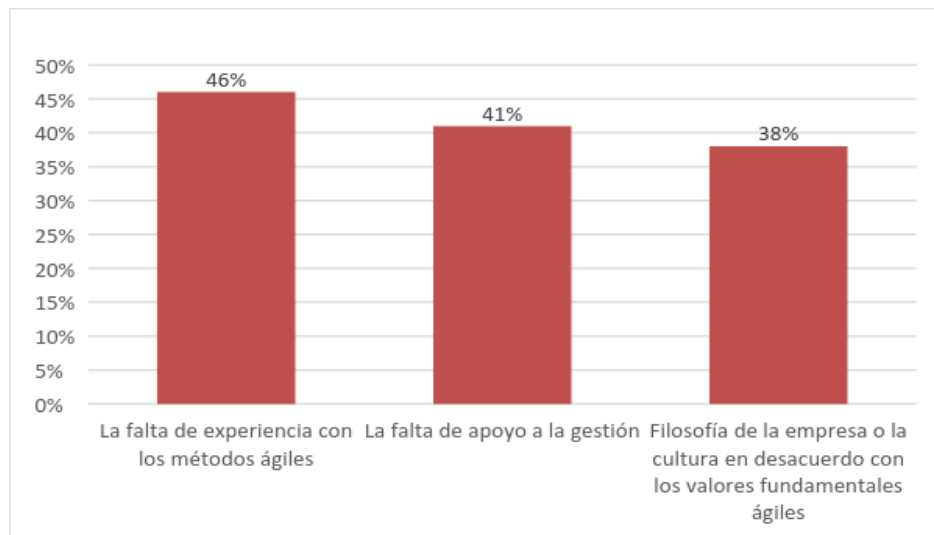


Figura 6. Causas que impiden adoptar las metodologías ágiles. (Version One, 2015)

En esta encuesta cuando se les preguntó los encuestados cómo miden el éxito de sus iniciativas ágiles, el indicador de valor más citado era del tiempo de entrega de proyectos 58%, seguido de la calidad del producto con el 48% y la satisfacción del cliente/usuario con el 46% como las principales punto de medición; como podemos analizar vemos una relación de “satisfacción de cliente = a tiempo la entrega del proyecto + producto con calidad”, como puntos determinantes de una metodología exitosa.

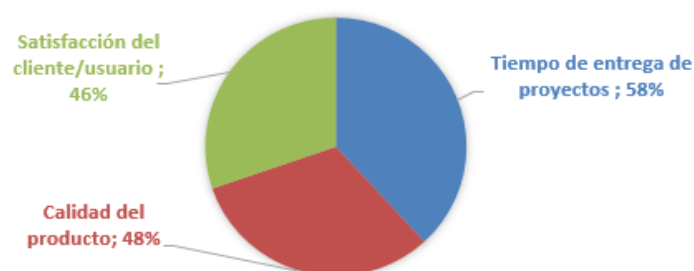


Figura 7. Perspectiva de cómo miden el éxito de sus iniciativas ágiles. (Version One, 2015)

3.2 Exploración de datos en el marco nacional

En relación al ámbito local se realizó una encuesta ver **Anexo A** basada en las preguntas de la encuesta realizada por Versionone, (2015). En la encuesta que se realizó a 38 desarrolladores de software de distintas empresas de desarrollo en la ciudad de Quito, determinó que un 50% de los encuestados utilizan el método SCRUM, seguido por el 23,7% que utilizan metodologías ágiles personalizadas y el 18,4% utiliza el método *eXtreme Programming*, como podemos observar la tendencia nacional e internacional (Figura 3) es usar la metodología SCRUM.

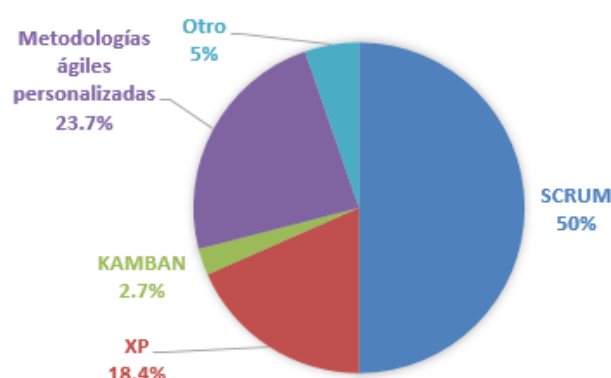


Figura 8. Uso de las metodologías ágiles (Datos locales). Elaboración propia.

En la pregunta cuáles fueron las mejoras que se evidenciaron al implementar metodologías ágiles, el 50% de los desarrolladores dijeron unas de las principales mejoras fue equipos productivos, un 63,2% se pronunciaron por la habilidad para gestionar cambios prioritario y el 34,2% para visibilidad del proyecto, como vemos las 3 mejoras significativas para el uso de metodologías ágiles con mayores votos son iguales en el mercado nacional como en el internacional.

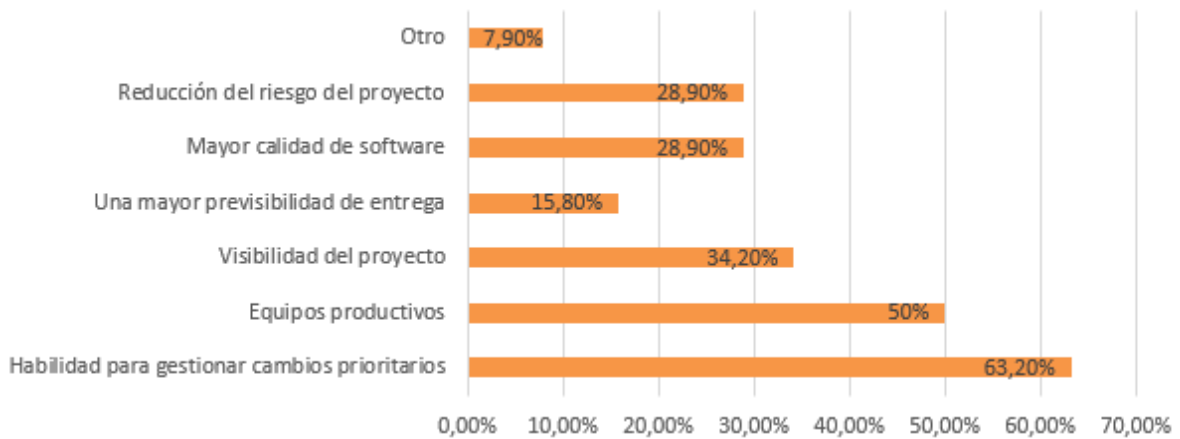


Figura 9. Las principales mejoras que se evidenciaron al implementar metodologías ágiles (Datos Locales).
Elaboración propia.

En la pregunta de causas que impiden adoptar la metodología ágil el 47,4% dijo que la resistencia al cambio es la principal causas que impide adoptar los métodos ágiles, mientras que el 39,5% de se manifestó por la opción “no existe personal con suficiente experiencia, dificultad sobre el cambio en la cultura organizativa” y el 34,2% por problemas en la gestión por falta de planificación”.

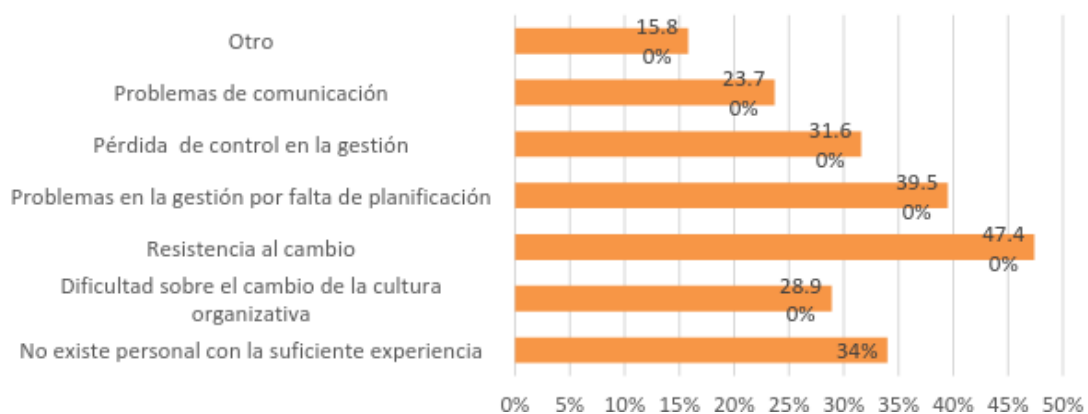


Figura 10. Causas que impiden adoptar las metodologías ágiles. Elaboración propia.

Cuando se les preguntó cómo miden el éxito de haber usado una metodología ágil, el 63,2% dijo que el tiempo de entrega del proyecto, el 57,9% se pronunció por la satisfacción con el cliente y el 44,7% por la productividad de los desarrolladores, como observamos en el contexto internacional y nacional son las misma 3 principales medidas con el que miden el éxito de las metodologías ágiles.

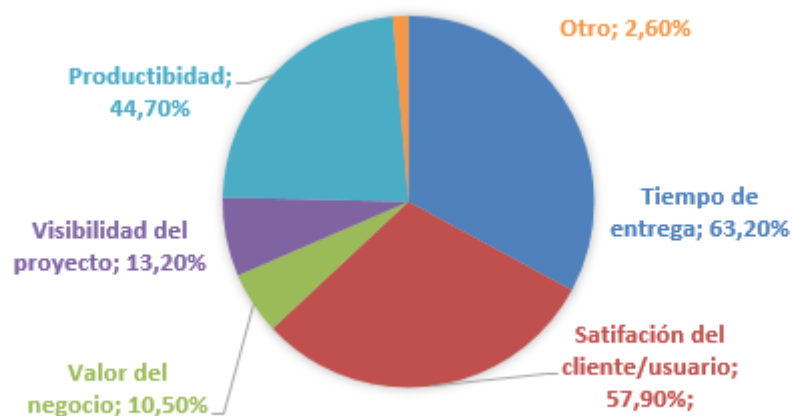


Figura 11. Perspectiva de cómo miden el éxito de sus iniciativas ágiles (Datos locales). Elaboración propia.

En la pregunta, cuáles son las técnicas de metodología ágil que más utiliza, el 52,6% respondió que la planificación de las iteraciones, el 42,1% dijo que utilizaba interacciones cortas, el 36,8% se manifestó por la opción de integración continua, haciendo un contraste con los datos internacionales solamente las planificaciones de las iteraciones se encuentra entre las técnicas más usadas en el ámbito local con respecto al internacional.

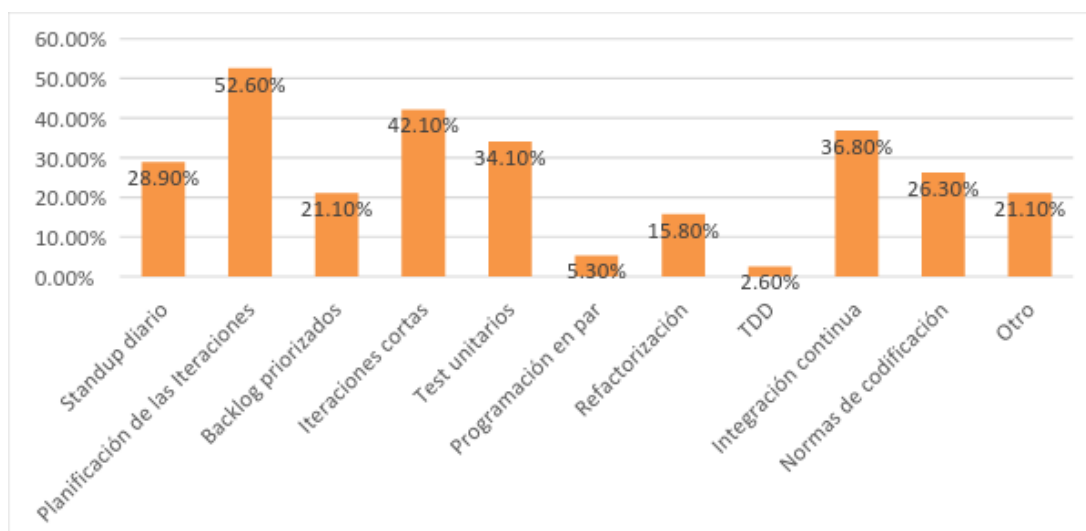


Figura 12. Técnicas usadas de metodologías de desarrollo ágil (Datos locales). Elaboración propia.

3.3 Elección de las metodologías a comparar

En función de los datos obtenidos, podemos decir que la metodología SCRUM y combinaciones de la misma es por largo la metodología más usada por los desarrolladores hoy en día. Otra tendencia es usar metodologías híbridas que no es más que una combinación de metodologías donde se toma las mejores prácticas de cada una de ellas tal como es el caso de Scrumban y Scrum/XP.

Las metodologías a analizar se seleccionaron por el nivel de uso y la cantidad de información que se encuentra disponible sobre cada una, evidenciadas en las encuestas realizadas a nivel internacional como local, ya que plasman el grado de influencia en el ambiente laboral en empresas de desarrollo de software.

En las entrevistas realizadas a profesionales ver **Anexo B** mencionan en el que la elección de la metodología a usar, depende en gran medida de las necesidades y características de la aplicación a construir, pero existe influencia en la utilización de metodología Scrum, XP, Kanban o combinación de estas; por otra parte las nuevas metodologías orientadas al desarrollo de aplicaciones móviles como las mencionadas Hybrid Methodology Design y Development Process Spiral no sean experimentado en gran medida o no existe suficiente recursos bibliográficos sobre estas metodologías por lo que no serán tomadas en cuenta para el desarrollo de este trabajo.

Las metodologías seleccionadas para la realización de este trabajo son: Scrum, Extreme Programming, Kanban y Mobile-D.

3.4 SCRUM

Los creadores de esta metodología la definen como “un marco de trabajo por el cual los individuos pueden embestir problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible, productiva y creativamente”(Schwaber & Sutherland, 2013, p.4).

Scrum es:

- Ligero
- Fácil de entender
- Extremadamente difícil de llegar a dominar

Esta metodología se centra en actividades de gestión de proyecto y no la especificación de prácticas de ingeniería; por lo que se puede emplear varias técnicas y proceso de ingeniería de software para la construcción del proyecto, fomentando los equipos auto dirigidos, cooperativo y teoría de control empírico de los procesos para el seguimiento e inspección del proyecto. Este método funciona bien cuando tiene objetivos o actividades con prioridad bien definidas y los requerimientos son inestables, dado que el mismo tipo de proyecto fomenta el surgimiento de estas ambigüedades (Peralta, 2003).

SCRUM tiene un enfoque en el conocimiento a diferencias de las metodologías tradicionales que definen la construcción de un proyecto de software como un “proceso definido”, ya que SCRUM procede de la experiencia para la toma de decisiones, empleando un enfoque repetitivo e incremental para mejorar la predictibilidad y el control del riesgo, por lo que es crucial tener desarrolladores con conocimientos sólidos dadas las características del desarrollo de aplicaciones móviles para el proyecto.

Esta metodología tiene una estrecha colaboración con el cliente dando como características (Schwaber & Sutherland, 2013, p.4):

- Predisposición y respuesta al cambio.
- Desarrollo iterativo e incremental con entregas frecuentes de funcionalidad del proyecto.
- Comunicación verbal directa.
- Simplicidad, sólo los artefactos necesarios.
- Motivación, compromiso y responsabilidad del equipo para la autogestión y auto organización.

Centrarse en la comunicación con un cliente específico puede generar problemas de diseño e iteración ya que en su marco de trabajo no visualiza las necesidades de usuarios objetivos.

3.4.1 Teoría del control empírico

El control empírico asume que siempre existen cambios debido a la complejidad e indeterminación del proyecto, se fundamenta en 3 pilares que son: transparencia, inspección y adaptación los cuales se señalan en “La guía de SCRUM” (Schwaber & Sutherland, 2013).

- Transparencia: Contempla los aspectos representativos del proceso que deben ser claros para aquellos que son encargados del resultado, lo cual requiere que dichos aspectos sean

definidos por un estándar común para que todos los involucrados tengan un entendimiento de lo que se está haciendo.

- Inspección: Se deben inspeccionar frecuentemente los artefactos de SCRUM, el encauzamiento hacia una meta, para detectar cambios. La inspección no debe interferir en el trabajo y deben ser realizadas por inspectores expertos.
- Adaptación: Si se determina que una característica de procedimiento se desvía de límites aceptables, y que el software consecuente no será admisible. Este proceso o característica deben ser ajustado cuanto antes para minimizar desviaciones mayores y evitar que el proyecto no sea aceptado.

3.4.2 Sprint

Es el bloque de tiempo en el cual se crea un incremento de software finalizado, disponible y potencialmente desplegable, cada nuevo sprint comienza al finalizar el sprint previo, un sprint está compuesto por los siguientes eventos: Reunión de Planificación del Sprint, Scrum Diario, Revisión del Sprint, Retrospectiva del Sprint. En relación a la creación de sprint con respecto a la construcción de una App, estos generan funcionalidades (módulos de la aplicación) independientes de sprint futuros lo cual puede acarrear problemas por no tener en cuenta características tecnológicas de hardware y software. (Schwaber & Sutherland, 2013)

El marco de trabajo Scrum está compuesto por equipos *SCRUM*, roles, eventos, artefactos y reglas asociadas.

3.4.2.1 Objetivo del sprint (Goal Sprint)

El objetivo del sprint es una meta que se establece en el sprint que se debe lograr mediante la dotación de la “Lista de Producto”, ofreciendo orientación de qué es lo que se desea en el incremento. (Schwaber & Sutherland, 2013)

3.4.2.2 Eventos formales

Los eventos formales están diseñados para evitar reuniones no definidas en *SCRUM* definidos en bloques de tiempo con una duración máxima y que pueden terminar siempre y cuando se haya alcanzado los objetivos del evento, los siguientes eventos son los definidos por SCRUM (Schwaber & Sutherland, 2013):

- Reunión de Planificación del Sprint (Sprint Planning Meeting)
- Scrum Diario (Daily Scrum)
- Revisión del Sprint (Sprint Review)
- Retrospectiva del Sprint (Sprint Retrospective)

3.4.2.3 Artefactos de SCRUM

Representan las formas útiles para proporcionar transparencia, inspección y adaptación del sprint en las que destaca la Lista de Producto (Product Backlog) el cual es una lista ordenada de los que es necesario en el producto, es única fuente de requisitos, esta lista enumera todas las características, funcionalidades, requisitos y correcciones que componen la creación de la iteración. (Schwaber & Sutherland, 2013)

3.4.2.4 El equipo SCRUM (SCRUM Team)

El grupo está integrado por el equipo de desarrollo (Development Team), el dueño del producto (Product Owner), y un SCRUM Master los cuales son auto organizados y multifuncionales de tal manera que el equipo sea totalmente independiente de personas externas, optimizando la flexibilidad, creatividad y la productividad generando versiones útiles y funcionales del producto dada la opulencia de dispositivos móviles y las múltiples plataformas que existen es recomendable que el equipo SCRUM tenga conocimientos firmes en la construcción. (Schwaber & Sutherland, 2013)

En la siguiente figura se presenta un esquema de trabajo SCRUM:



Figura 13. Esquema de trabajo SCRUM. (Deemer, Benefield, Bas, & Versión, 2009)

3.5 KAMBAN

Kamban es una metodología de producción de trabajo continuo y sin sobrecarga en el equipo de producción innecesario en stock, Anderson y Reinertsen (2010) lo que significa que la producción de tareas se desarrollan una a la vez y tomando tareas prioritarias; por lo tanto concede identificar las obstrucciones en la producción, acrecentar los tiempos de servicio y calidad de los procesos.

Kamban requiere una alta madurez y profesionalidad dado que es un sistema de trabajo WIP² “(work in progress)” (Bermejo, n.d.) quitando los sobrantes, demostrando que cuanto más trabajo existe, más baja es la calidad. Este postulado hace referencia a que los desarrolladores realizan un mejor trabajo si sólo se enfocan en una tarea a la vez, ya que mejoran la velocidad de respuesta y la calidad con la que se construye la respuesta. Generando así una relación inversamente proporcional entre cantidad de trabajo paralelos y la calidad y velocidad de desarrollo.

Dado que es una metodología se centra en cómo limitar el número de tareas que se deben realizar muchos autores coincide en la siguiente fórmula:

$$\text{Límite WIP} \leq n^{\circ} \text{ de tareas en curso} \leq n^{\circ} \text{ de tareas hechas pendientes. (Bermejo, n. d.)}$$

La palabra Kamban viene del japonés “kan”=“visual” y “ban”= “tarjeta o tablero”, lo cual significa tarjeta visual el cual hace referencia al panel o tablero Kamban el cual es continuo y en el que las nuevas tareas se acumulan en la sección inicial y en el cada columna representa las fases con los números límites de trabajos (Gimson, 2015).

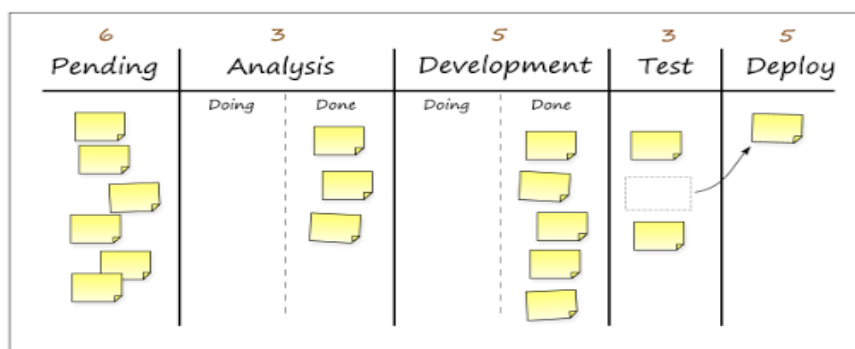


Figura 14. Tablero KANBAN. (Broullón, 2013)

² El WIP es una técnica de manufacturación propia de la metodología de procesos de producción KAMBAM creada por la empresa TOYOTA, esta a su vez fue elaborada a partir de la metodología LEAN la cual utiliza la técnica JIT (Just in time); <http://www.javiergarzas.com/2011/11/kanban.html>.

Kanban trata de hacer un balance entre la capacidad de trabajo y priorizar las tareas que mejoren el valor del negocio.

3.5.1 Multitareas

El autor Bermejo, nos explica con el ejemplo:

“En las autopistas, a medida que aumenta la densidad de tráfico, la velocidad del tráfico disminuye. A priori, esta disminución de velocidad no tendría que ocurrir porque, si todos los coches fueran a la misma velocidad, no habría motivo para que el tráfico fuera lento o incluso se detuviera. El problema viene cuando los coches cambian de carril. Un cambio de carril hace que el coche que cambia reduzca la velocidad, por lo tanto, el de detrás también y esto se propaga hasta reducir considerablemente la velocidad de todo el flujo de coches completo”.(Bermejo, n.d., p.16)

Como el hecho de realizar multitareas afecta negativamente el tiempo de servicio y el de la producción en general, puesto que el cambio de tarea implica tomarse el tiempo para entender los nuevos requerimientos, además que entre mayor multitareas haya, más lenta será la resolución del problema generando así un problemas de desperdicio de tiempo..



Figura 15. Efecto del desarrollo de tareas paralelas.(Bermejo, n.d.)

Como nos sugiere (Weinberg, 1992) en el que indica que se malgasta un 20% del tiempo por cada tarea adicional que nos adjudicamos a la vez. En Kanban el trabajo es conseguir que las tareas fluyan por el panel lo más ágil posible, minimizando el tiempo que una tarea se queda en una fase.

Esta metodología dicta tener siempre un “trabajador T” (Bermejo, n.d.), que es el que conoce y se interesa en todas las etapas del proyecto albergando líneas verticales y horizontales del conocimiento sobre una ciencia, esta persona es la que ayuda y agiliza los cuellos de botellas que se podrían crear generalmente en las etapas finales.

Los tres fundamentos básicos del Kamban son los siguientes:

- 1) Visualizar de forma continua el estado del equipo de producción,
- 2) Limitar el WIP para mejorar la calidad y el tiempo de entrega, y
- 3) Potenciar el flujo.

Por lo que esta metodología nos ayuda a: “evita los olvidos, mala información, malos entendidos, duplicación, pérdida de calidad y retrasos. Los límites deben ser fijados empíricamente, con prueba y error” (Gimson, 2015, p.37).

Kanban es una metodología simple que basa en: Visualización del flujo de trabajo, Limitación del trabajo en proceso, Medición y Gestión del Flujo y Mejora incremental. Como lo refieren los creadores de esta metodología, son justamente los métodos simples los que generan los menores conflictos y generalmente los más beneficiosos (Gimson, 2015).

3.5.2 Funcionamiento

Esta metodología puede visualizarse como un sistema transparente y visual de limitación del trabajo en curso y arrastre (*pull*) del valor, pero el cual puede representar algunos problemas como el hecho de que una persona haya terminado su trabajo y no tenga que arrastrar de la etapa anterior, dentro de las normativas de Kamban este estable que el recurso libre (personas) deben ayudar al compañero siempre sea posible. (Kniberg & Skarin, 2010)

3.6 XP (EXTREME PROGRAMMING)

Es una metodología centrada en los principios de la agilidad tales como desarrollo incremental, relación activa con el cliente, el interés en las personas y no en los procesos, retroalimentación, simplicidad en las soluciones implementadas y afrontar los cambios que puedan surgir. Los autores Letelier & Penadés (2006) definen a XP como “especialmente

adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico”, estas características lo hace conveniente ya que el desarrollo de aplicaciones móviles es muy complejo.

XP tiene como objetivos según (Oiver & Pérez, 2011) “... la satisfacción del cliente y potenciar al máximo el trabajo en grupo...”, en el cual se atiende las necesidades del cliente de una manera rápida, potenciando el esfuerzo del grupo de desarrollo en generar las soluciones que se requieren, para la realización de una app se debe considerar que el cliente viene siendo un grupo de personas objetivos por lo que los esfuerzo en la construcción de la aplicación es tratar de prever la necesidades de todo este conjunto de personas.

3.6.1 Prácticas

XP se fomenta en 10 prácticas que involucran al equipo de trabajo, los procesos y el cliente (Joskowicz, 2008):

- **Planificación Integral:** Es la toma de requerimiento la cual se genera entre el cliente y el equipo de desarrollo y son negociadas progresivamente.
- **Entregas pequeñas:** En la cual se desarrolla las funcionalidades mínimas que permitan la utilización lo antes posible y en el cual se irá completando gradualmente.
- **Diseño sencillo:** Se debe mantener siempre un código simple, no redundante.
- **Desarrollo previamente aprobado:** En esta práctica se hace referencia a primero escribir las pruebas de validación y después el código esto con la finalidad de asegurar la calidad de software.
- **Limpieza del código o refactorización:** Consiste en simplificar y optimizar el producto sin perder funcionalidad, en cada nuevo requerimiento que se integra al software.
- **Programación en Pareja:** En el cual la codificación de las funcionalidades se realiza en pares, en el cual uno piensa en la solución y codifica; y la otra persona piensa estratégicamente cuestionando y replanteando el problema generando un *feedback* de la solución que se está aplicando.
- **Propiedad colectiva:** Esta práctica establece que cualquier elemento del equipo de desarrollo puede entender el código y por lo tanto modificar cualquier parte del código, dado que se desarrolló la solución mediante programación por pares.

- Integración continua: Esta práctica dicta integrar al sistema entero la nueva funcionalidad y generar las pruebas a todo el sistema lo cual busca garantizar la funcionalidad del todo el software con el nuevo requerimiento añadido.
- Ritmo sostenible: No trabajar en cantidades superiores a 40 horas semanales dado que según (González Campos y Fernández Martínez, 2015) “la velocidad máxima a la que se debe trabajar es 8 horas diarias”.
- Cliente presente: En el cual se establece que el cliente o un representante de este debe estar presente para contestar las interrogantes que puedan surgir.

3.6.2 Fases de XP

Las fases en el marco de trabajo de XP son las siguientes (Joskowicz, 2008) (Echeverry Tobón & Delgado Carmona, 2007):

Fase de Planeación

Es la etapa inicial de proyectos, la cual inicia con las historias de usuarios donde se describen las características y funcionalidades del software, con el cual se crea un “*release planning*” donde se publicarán las fechas de las versiones que se crearan dado el tiempo en el que se implementara en la realización de las historias de usuarios. También se genera un plan de iteraciones y se especifican cuáles serán los criterios de pruebas.

Fase de Diseño

El proceso de diseño debe proyectar diseños simples, sencillos y puntuales, para facilitar el desarrollo. Este proceso se apoya en el uso de tarjetas CRC (Class, Responsibilities and Collaboration) las cuales identifican las clases orientadas a objetos con mayor relevancia. Además de especificar los nombres de las clases y métodos para comprender el diseño.

Fase de codificación o desarrollo

En esta fase el cliente es una parte más del equipo de desarrollo dado que cliente debe especificar a detalle lo que la historia de uso debe hacer, primero se deben realizar las pruebas de unidad que deben pasar y después realizar la solución de la funcionalidad que debe pasar la prueba.

Fase de Pruebas

En esta fase las se aplican las pruebas en todo el proyecto después de haber añadido la nueva funcionalidad al proyecto, con se espera encontrar en el caso que hubiera errores de una forma más rápida y a tiempo, en esta fase se realizan pruebas de unidad y de aceptación. Solo se tendrá una nueva versión se las pruebas en su totalidad han sido pasadas (Echeverry Tobón & Delgado Carmona, 2007).

3.6.3 Roles

En la metodología XP se establecen características claras de cada uno de los roles (Echeverry Tobón & Delgado Carmona, 2007):

- Programador: Es el encargado de codificar las pruebas unitarias y solución de las historias de usuarios.
- Cliente: Se encarga de generar y priorizar las historias de usuarios, diseña las pruebas de aceptación y siempre debe estar disponible para cualquier interrogante.
- Encargado de pruebas (Tester): Ayuda a diseñar las pruebas de aceptación, a ejecutar y automatizar las pruebas de aceptación e integración y difunde los resultados al equipo de desarrollo.
- Encargado de seguimiento (Tracker): Se encarga de realizar el *feedback* de todo el proceso XP, evaluando las estimaciones y resultados generados en el proceso.

La siguiente figura muestra el esquema de trabajo de XP:

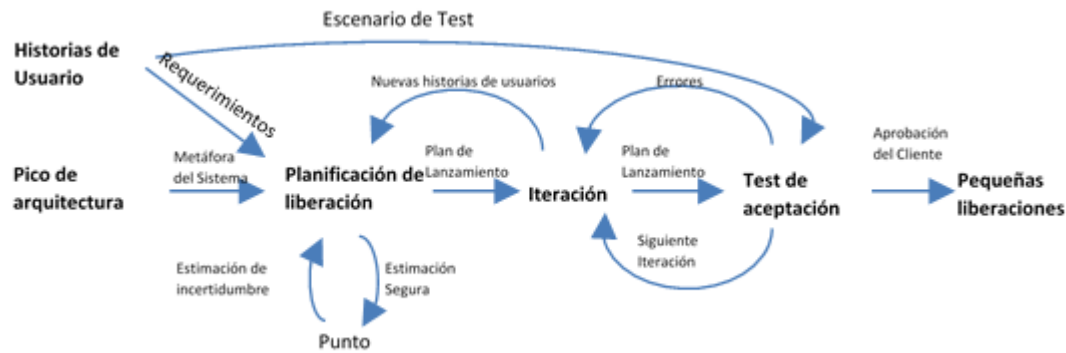


Figura 16. Esquema de las fases de XP. (Donvan Well, 2000)

La siguiente figura muestra el modelo de iteraciones en la metodología XP:



Figura 17. Modelo de Iteraciones en XP. (Donvan Well, 2000)

3.7 Mobile - D

Mobile-D es una metodología ágil orientada al desarrollo de aplicaciones móviles desarrollado por Pekka Abrahamsson y su equipo de trabajo del VTT (Valtion Teknillinen Tutkimuskeskus, en inglés Technical Research Centre of Finland) en Finlandia, orientados a ciclos de desarrollos muy rápidos y grupos pequeños de desarrolladores, máximo 10 personas.

Mobile-D se basa en las prácticas de eXtreme Programming (XP), metodología Crystal para la escalabilidad de los métodos y Rational Unified Process para el diseño completo de ciclo de vida (Blanco, Camarero, Fumero, Warterski, & Rodriguez, 2009). Se basa en los nueve elementos principales que envuelven a todas las fases de desarrollo (Abrahamsson & VVT, 2010), las cuales son:

- **Fases y colocación**

Cada una de las iteraciones del proyecto comienza con un día de planificación.

- **Línea de Arquitectura**

El enfoque de línea de la arquitectura se utiliza para capturar el conocimiento de las fuentes existentes para generar soluciones arquitectónicas cuando sea necesario.

- **Test Driven Development móvil**

Los ensayos primera aproximación se realizan junto con los casos de prueba automatizados.

- **Integración continua**

Las prácticas efectivas de SCM (Gestión de cadena de suministros, “*Supply Chain Management*”) se aplican a través de múltiples medios.

- **Programación Par**

Codificación, pruebas y *refactoring* se llevan a cabo en pares.

- **Métricas**

Se establece pocas métricas esenciales y son utilizados para los propósitos de mejora de retroalimentación y de procesos.

- **Mejora de Procesos de Software Ágil**

Los talleres post-iteración se utilizan para mejorar continuamente el proceso de desarrollo.

- **Fuera de las instalaciones del cliente**

Crear participación del cliente en la planificación y la liberación de la aplicación.

- **Enfoque centrado en el usuario**

Se hace hincapié en la identificación y satisfacción de las necesidades de los usuarios finales.

3.7.1 Fases

Esta metodología está compuesta por 5 fases y cada una estas se asocian a etapas, tareas y prácticas, las cuales están especificadas en el documento oficial.



Figura 18. Fases y etapas de Mobile-D.(P. Abrahamsson & VVT, 2010)

Fase de Exploración

Se establece la planificación y características del proyecto, es una fase importante ya que establece las bases para la aplicación, como el proceso de la aplicación, ambiente, arquitectura. En esta fase se necesita a las personas interesadas para proporcionar su experiencia.

Fase de Inicialización

En esta fase se estable los recursos físicos, técnicos, humanos y de comunicación; así como todas las partes activas que comprende el producto

Fase de Producción

En esta fase se implementa las funcionalidades requeridas en el producto mediante un ciclo de desarrollo iterativo e incremental.

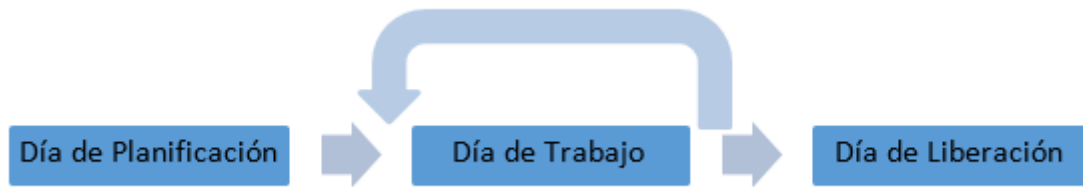


Figura 19. Desarrollo iterativo e incremental en la fase de producción. (Abrahamsson & VVT, 2010)

Fase de Estabilización

El propósito de esta fase es garantizar la calidad del producto, el cual es presentado al *stakeholder*; todas las fuerzas se centrarán la integración del producto

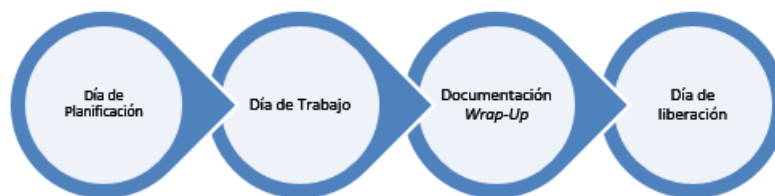


Figura 20. Fase de Estabilización. (Abrahamsson & VVT, 2010)

Fase de Pruebas y arreglo del sistema

El propósito esta fase es ver las funcionalidades del producto en producción definidas por el cliente correctamente, proporcionando *feedback* al equipo sobre el proyecto y corregir defectos encontrados.

Los autores de esta metodología dicen haberlo probado, obteniendo una certificación CMMI (Capability Maturity Model Integration)³ 12 de nivel 2. Esto parece ser una ventaja competitiva ante las otras metodologías, “puesto que la contratación de empresas para la externalización del desarrollo software se rige por la auditoría de los ciclos y técnicas de desarrollo que utilizan” (Blanco et al., 2009, p.15).

3.8 Análisis de las metodologías

³ CMMI, es una de las métricas de aseguramiento de calidad, para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación más aceptados en el sector tecnológico. <https://www.sei.cmu.edu/cmmi/>

Dado las características del desarrollo de aplicativos móviles como la identificación de los actores (quien va a utilizar el producto) las metodologías como XP, Kanban y Scrum necesitan que el cliente tenga una participación activa en el ciclo de vida de cada una, sin embargo no consideran en el desarrollo de este, un enfoque a los usuarios finales y la usabilidad que son los principal actores del producto como la metodología Mobile-D, dándole a este una perspectiva diferente a la construcción del aplicativo, cabe destacar que Mobile-D no deja de lado la interacción con el cliente.

Además, a pesar que las metodologías ágiles dicta que la integración del cliente en el ciclo de vida de las metodologías es primordial, se da en la mayoría de los casos que existe una ausencia del cliente en etapas posteriores a la de planificación, generando riesgos en la realización de la aplicación. En este punto los valores o principios de las metodologías como la orientación para la utilización de un grupo de personas objetivos y no prescindir por completo del cliente es totalmente pertinente, de la cual XP y Mobile-D hacen eco de esta característica.

Otra característica importante es que la aplicación gane mercado, por lo que el tiempo de entrega de las versiones de la aplicación (release) debe ser corto, todas las metodologías ágiles nos permiten realizar entregables en tiempos considerablemente corto pero Kanban al ser un sistema WIP (*Work In Progress*) nos permite comprometer los SLA (Acuerdos de Nivel de Servicio, en inglés Service-Level Agreements)⁴ y hacer entrega realistas en tiempos más bajo que las otras tres metodologías, aunque estas se puedan acoplar a entrega de versiones funcionales para los usuarios finales.

Dada la singularidad de las limitaciones inherentes y evolutivas de los dispositivos móviles citadas con anterioridad todas manejan fases planificación, de integración continua y pruebas unitarias con la finalidad de generar un producto de calidad, pero solo Mobile-D plantea una fase estabilización vital en el desarrollo de aplicativos móviles dado la variedad de dispositivos móviles que existen hoy en día, así como un enfoque hacia la arquitectura y ambiente que debe tener el software.

⁴ Un SLA es un acuerdo o contrato entre el proveedor y el cliente, en la que se describen los servicios, especifican responsabilidades, objetivos, estipulado entre ambas partes y el cual refleja contractualmente un nivel operativo de funcionamiento exponiendo un nivel mínimo de aceptación de servicio para el cliente. (<http://www.contratosinformaticos.com/sla/>).

Los SLA son un modelo de acuerdo de servicio la cual es parte de la Gestión de los Niveles de Servicios definidos en el proceso de diseño del marco de trabajo para la gestión de servicios de tecnologías de la información ITIL. (<http://itilv3.osiatiss.es/itil.php>).

El estudio realizado por (Flora & Chande, 2013) se introduce en la metodología Mobile-D un enfoque de desarrollo de software dibujado desde el campo de la ingeniería de software ágil, que está diseñado para satisfacer las demandas específicas de entorno móvil extremadamente volátil el cual lo describe como vital y la cual no es parte de los principios de las metodologías XP, Scrum y Kanban.

En las metodologías ágiles un factor sin mucha relevancia es la documentación generada pero que sin duda en un ambiente cambiante como es el desarrollo de aplicaciones móviles es imprescindible, en metodologías como SCRUM y XP, la documentación no marca el ciclo de vida pero existe artefactos o tareas que apoyan esta necesidad como CRC en el caso de XP e historias de usuarios en el caso de SCRUM, no obstante la metodología MOBILE-D para solventar esta necesidad plantea tareas de documentación (Wrap-Up) en la fase de estabilización.

Otra particularidad deseada en las metodologías ágiles para las aplicaciones móviles es la conciencia de producto en el mercado, es decir la perspectiva empresarial ya que las aplicaciones móviles además de ser un proyecto es un producto (Pekka Abrahamsson, 2005) por lo que deben integrar prácticas de diseño de nuevos productos en el que integren reducción del time-to-market⁵.

Y como parte esencial en el desarrollo de software, la fiabilidad en los resultados “obtener lo que se desea” en un ambiente tan cambiante; XP, Mobile-D y SCRUM manejan enfoques de construcción adaptivos en respuesta a los requisitos cambiantes que salen a flote, como característica perceptible se tiene la realización de planificaciones para la resolución de nuevas necesidades o errores reportados en la que metodologías como XP, mientras que en Mobile-D se realizan según las condiciones que lo ameritan, ya que no se necesita la culminación de objetivos propuesto con anterioridad como el caso de Scrum.

Por otra parte, la adaptación de cada una de las metodologías dependerá de las necesidades de la aplicación en las que es necesario la experiencia para poder dictaminar el “cómo, cuál y qué aplicar” de cada una de las metodologías y definir el flujo o marco de trabajo con el que se trabajará.

Por lo que, en la problemática actual que existe en el desarrollo de aplicaciones móviles, es crucial una metodología de requisitos y una estrategia de diseño que haga frente a esta

⁵ (“Time to Market - Accenture,” n.d.) El Time to market se define como la capacidad de reacción que tienen las organizaciones para crear o mantener ventajas competitivas ante los retos que presenta el mercado y sus competidores.

problemática y poder desarrollar un producto eficiente, funcional y seguro. En la siguiente tabla podemos analizar en base a las características que requiere el desarrollo de aplicaciones móviles podemos determinar que por sus características Mobile-D tiene mejores cualidades para este tipo de desarrollo cabe destacar que entre las metodologías citadas es la única que enfoca sus principios en ese ámbito de desarrollo de software.

Tabla 3. Cuadro de comparaciones de metodologías ágiles. Elaboración propia.

Características para el desarrollo de aplicaciones móviles	SCRUM	KAMBAN	EXTREME PROGRAMMING	MOBILE - D
1 Identificación de Usuarios Objetivos/ Usuarios Finales.	Enfocado en las necesidades de cliente.	Enfocado en las necesidades de cliente.	Enfocado en las necesidades de cliente.	Enfoque en los usuarios finales.
2 Comunicación.	Comunicación alta.	Comunicación alta.	Comunicación alta.	Comunicación muy alta.
3 Documentación.	Documentación media.	Documentación baja.	Documentación media.	Documentación producida alta.
4 Minimización de impacto de limitaciones inherentes.	Alto.	Baja.	Alta.	Muy alta.
5 Minimización de impacto de limitaciones de evolución.	Alta.	Baja.	Alta.	Alta.
6 Valores o principios orientados al desarrollo de App.	Principios orientados a la gestión de proyectos.	Principios orientados a potenciar el flujo de trabajo.	Principios orientados a potenciamiento del trabajo del grupo de desarrolladores.	Elementos orientados al desarrollo de App.
7 Entrega de la aplicación en tiempos cortos.	Tiempos de entrega largos.	Tiempos de entrega cortos.	Entregas pequeñas (funcionalidades).	Entregas pequeñas (funcionalidades).
8 Metodología fácil de adoptar.	Dificultad muy alta.	Dificultad media.	Dificultad media.	Dificultad alta.
9 Calidad del producto.	Calidad alta.	Calidad alta.	Calidad alta.	Calidad alta.
10 Conciencia del producto en el mercado.	Media.	Baja.	Media.	Alta.

El siguiente cuadro de ponderaciones tiene valores que van desde uno a diez siendo uno el valor más bajo y diez el valor más alto y las cuales son otorgadas en función de las características y enfoques de las metodologías y experiencias de uso de las personas encuestadas, en la tabla 4 se define las característica con el mismo nivel de importancia.

Tabla 4. Tabla de ponderaciones de las metodologías ágiles vs características para el desarrollo de aplicaciones móviles. Elaboración propia.

Características para el desarrollo de aplicaciones móviles		SCRUM	KAMBAN	EXTREME PROGRAM MING	MOBILE - D
1	Identificación de Usuarios Objetivos/ Usuarios Finales	8	8	8	10
2	Comunicación.	9	9	9	9
3	Documentación	8	7	7	10
4	Minimización de impacto de limitaciones inherentes	8	6	8	9
5	Minimización de impacto de limitaciones de evolución	8	6	8	9
6	Valores o principios orientados al desarrollo de app	7	6	7	9
7	Entrega de la aplicación en tiempos cortos	8	9	10	10
8	Metodología fácil de adoptar	7	8	8	8
9	Calidad del producto.	9	8	8	9
10	Conciencia del producto en el mercado.	8	8	8	8
Total		80	75	81	91

Como se puede evidenciar a partir de la tabla 3 y 4 las características determinante que hacen “ganadora” a la metodología MOBILE-D es la identificación de usuarios finales, la generación de documentación, los valores o principios orientados al desarrollo de aplicaciones móviles y la entrega de la aplicación en tiempos cortos, debido a que se genera la necesidad propia de los proyectos móviles de que se tenga en cuenta a los usuarios finales ya que de eso

depende la usabilidad y experiencia de usuario de las aplicaciones, teniendo como involucrados directos e inseparables en la generación del proyecto la integración de los usuarios finales, clientes y desarrolladores.

Por otro lado la naturaleza cambiante de las tecnologías móviles hacen imprescindible la generación de documentación oportuna y necesaria para el mantenimiento de la infraestructura de la aplicación, aunque como principio de metodología ágil se tiene que no debe haber documentación exhaustiva, en el caso de los proyectos de para aplicaciones móviles generalmente entran en juego muchos componentes para la solución deseada, tal es el caso que la metodología MOBILE-D tiene como principio fundamental la generación de la línea arquitectónica para el desarrollo de la solución y de esta manera asociando un ciclo de vida de la metodología a la construcción de aplicaciones móviles, estos valores son los que hace énfasis la metodología antes mencionada y que las otras metodologías propuesta en este trabajo no.

Una característica importante y que no toma en cuenta ninguna de las metodologías propuestas en este trabajo, es un enfoque de diseño de nuevo producto (NPD por sus siglas en ingles), en el cual se analice el mercado objetivo, planes de comercialización del producto y cómo la aplicación móvil debe adherirse al modelo de negocio del cliente (empresa) y el cual debe generar valor para el cliente mediante indicadores de proyecto tales como calidad, gastos, riesgo, sostenibilidad y velocidad de obtención de objetivos las cuales forma el retorno de inversión de los proyectos (OBS Business School, n.d.).

Metodología Seleccionada

Manifestado los resultados, se determinó que teóricamente la mejor metodología para desarrollo de aplicaciones móviles entre las propuesta en este proyecto es Mobile-D por lo que la metodología a utilizar en el desarrollo del prototipo de aplicación será la antes mencionada.

3.9 Implementación de la Metodología en la construcción del prototipo del aplicativo móvil

Para la realización del prototipo de la aplicación móvil se implementará la metodología Mobile-D, en la cual se seguirá las fases y tareas establecidas en el sitio oficial de esta metodología⁶ y se implementará según las necesidades del caso.

3.9.1 Fase de Exploración

Como ya se trató en la sesión 3.7.1 en esta fase se definió y se estableció el stakeholder (personas involucradas en el proyecto), los requerimientos y alcance del proyecto, limitaciones y métricas de evaluación, el cual nos da una primera vista del proyecto con esto se pretende dar un adelanto de componentes y comportamiento de la solución a desarrollar y con la cual nos muestra un panorama de la línea arquitectónica que se va a plantear.

Modelo de Negocio de la Empresa

La empresa ofrece el servicio de rastreo y localización de las mascotas mediante la venta de dispositivos localizadores el cual utilizará la aplicación móvil para la interacción entre el usuario y la mascota.

Establecimiento de Stakeholder:

- Gerente de proyecto: Es el responsable de la planificación del proyecto, dar solución a problemas, comunicación con el team y cliente del proyecto.
- Team: El equipo multidisciplinario compuesto por jefe de proyecto, desarrolladores, diseñadores, arquitectos, tester, el cual es un equipo organizado y será el encargado de la construcción del producto.
- Cliente: Es la persona quien define el resultado, canaliza las necesidades del negocio, transmitiendo objetivos de valor al producto.
- Usuarios finales: Las personas quienes van a percibir el producto, dueño de mascotas que tienen la necesidad de saber dónde se encuentra su mascota.

⁶ Sitio oficial de la metodología MOBILE-D (P. Abrahamsson & VVT, 2010): <http://agile.vtt.fi/mobiled.html>

Definición del alcance del proyecto (Requerimientos iniciales):

- Una App que pueda localizar mascotas perdidas.
- El producto debe ser una App que identifique con un punto en el mapa dónde se encuentra la mascota.
- La App se debe comunicar con un localizador que debe tener la mascota.
- Debe ser de fácil manejo e intuitiva, someterse a métricas de diseño.
- La App se debe ejecutar en la plataforma donde se concentre la mayor parte de usuarios.
- La aplicación móvil deberá ser un prototipo funcional que satisfice los requerimientos iniciales.

Problemas que se puede encontrar en la construcción de la App:

- Posibles problemas que pueden surgir en el manejo de SMS desde la aplicación, por lo que se recomienda realizar la app en lenguaje nativo.
- La vía de comunicación entre el dispositivo GPS y la aplicación móvil es el sistema de comunicación móvil SMS (Short Message Service), por lo que la comunicación entre los dos puntos antes mencionados no es inmediata.
- Posibilidad de no poder gestionar números autorizados a los que debe responder el dispositivo.
- El sistema de posicionamiento GPS del dispositivo (Pet-Tracker) sólo funciona en ambientes abiertos por lo que genera una limitante en el funcionamiento de la aplicación.
- Para la utilización de la aplicación móvil primero el dispositivo de geo localización debe ser configurado por un técnico o especialista para un correcto funcionamiento.

Métricas para evaluar el proyecto:

- Velocidad de comunicación.
- UX (Experiencia de Usuario) Métricas de diseño de Material Designer. **(Ver Anexo F)**
- Satisface los requerimientos.
- Navegabilidad.
- Rendimiento de la App.
- Diversidad de dispositivos donde se ejecutará la aplicación.
- Estandarización del código PSR2 **(Ver Anexo F)** para PHP y Java Standar para Android **(Ver Anexo F)**.

3.9.2 Fase de Inicialización

En esta fase se hace tangible la línea base de la arquitectura para la solución de la aplicación móvil, así como un panorama de las vistas (Front) de la aplicación móvil todo con la finalidad de obtener alta calidad minimizando el impacto en las limitaciones inherentes y evolutivas tratadas en la sección 1.3.1 de este trabajo.

Establecimiento del proyecto – Definición de la línea base.

Dada las necesidades del manejo del broadcast para la comunicación de la aplicación móvil y el dispositivo de localización de la mascota se ha optado por el desarrollo en lenguaje nativo, además de la necesidad de desarrollar en lenguaje de código abierto y la gran variedad de documentación existente se ha escogido el IDE (entorno de desarrollo integrado) Android Studio, además dada la necesidad de comunicación a servicios orientados a internet para la transferencia de recursos se decide realizar una APIs RESTful (**Ver Anexo F**) dadas las ventajas de escalabilidad y rendimiento de este tipo de arquitectura ya que no se necesita del mantenimiento en memoria de sesiones, obteniendo así una independencia de tecnologías a utilizar.

Componentes de desarrollo:

Entorno de desarrollo de la App a configurar.

- Android Studio IDE.
- Android SDK Manager.
- Android Virtual Device AVD.

Entorno de desarrollo de la API RESTful a configurar.

- Vagrant para la creación de entornos virtuales de desarrollo.
- Homestead (Máquina Virtual de vagrant para el desarrollo de proyecto con el framework Laravel).
- Laravel Framework 5.2 de PHP.
- Servidor web Apache.
- Lenguaje de programación PHP versión 7.
- Motor de Base de datos MySQL versión 5.7

Plataformas de despliegue:

- Amazon Web Service, servicio EC2 (Elastic Compute Cloud) instancia t2.micro para la API RESTful y el servicio RDS (Relational Database Service) instancia del motor de base de datos MySQL.
- Google Play.

Herramienta para la gestión del proyecto a configurar:

- Trello (Gestor de tareas que permite el trabajo de forma colaborativa mediante tableros)
- Git (Software para el control de versiones)
- SourceTree (Es una aplicación cliente para Windows que proporciona la interfaz gráfica para el uso de Git)
- Bitbucket (Es un servicio de almacenamiento de proyectos basado en web que utilizan sistema de control de versiones)
- Archivo SPECS (Denominación a los archivos donde se especifican o documentan los endpoints de comunicación entre la aplicación móvil y el API RESTful)

Reseña de las herramientas utilizadas para la creación del prototipo de la aplicación móvil ver el **Anexo F** y en el cual están los enlaces hacia las páginas oficiales de las herramientas.

Arquitectura de la solución

El diagrama muestra la estructura, funcionamiento y comunicación entre la aplicación móvil, el dispositivo de rastreo de la mascota y el servidor web.

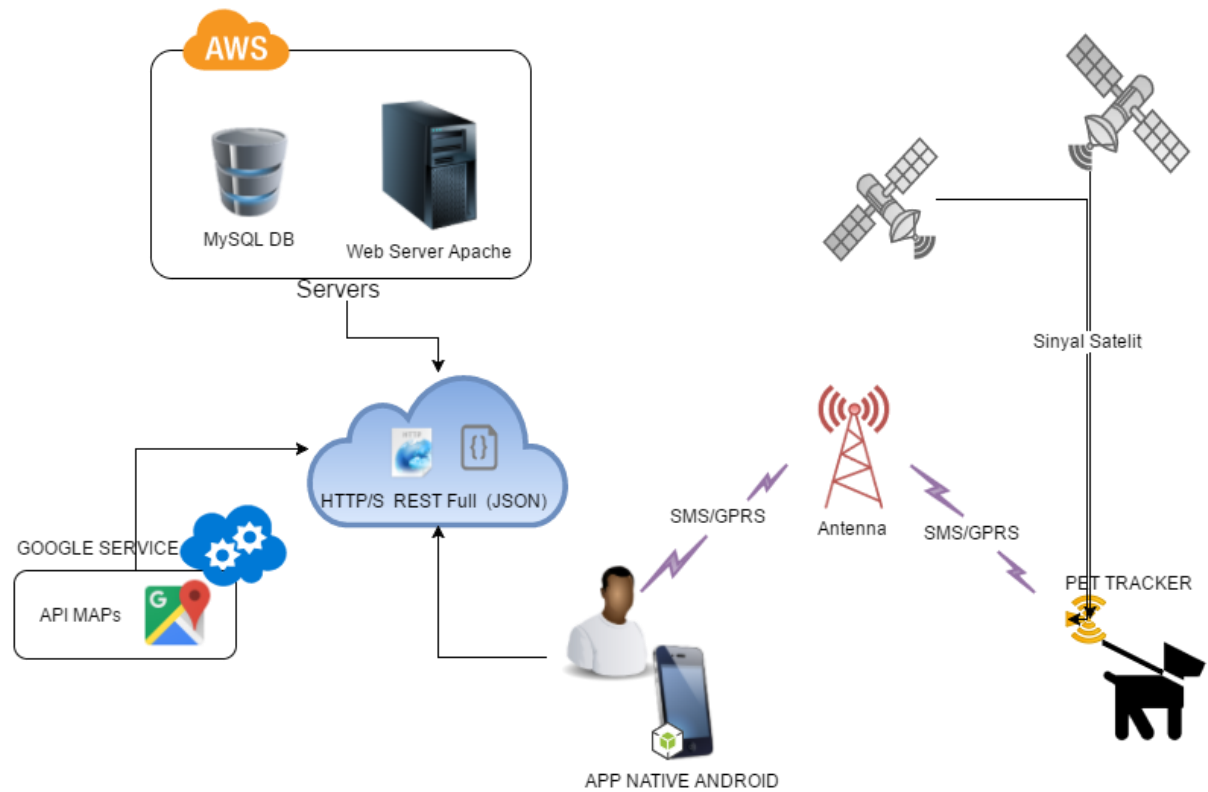


Figura 21. Arquitectura del sistema. Elaboración propia.

En las siguientes figuras se presenta de manera más detallada la arquitectura del servidor donde se encuentra alojado la API RESTful y la arquitectura de la aplicación móvil.

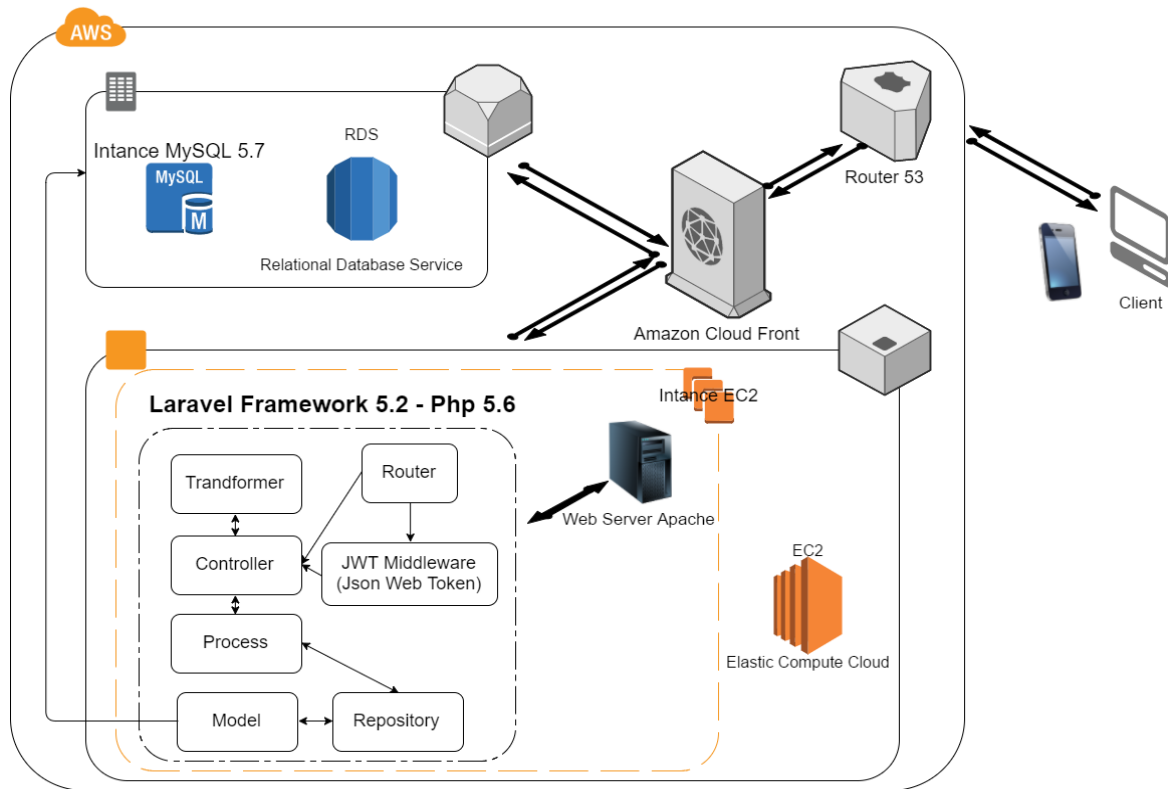


Figura 22. Arquitectura del Servidor Web – API RESTful. Elaboración propia.

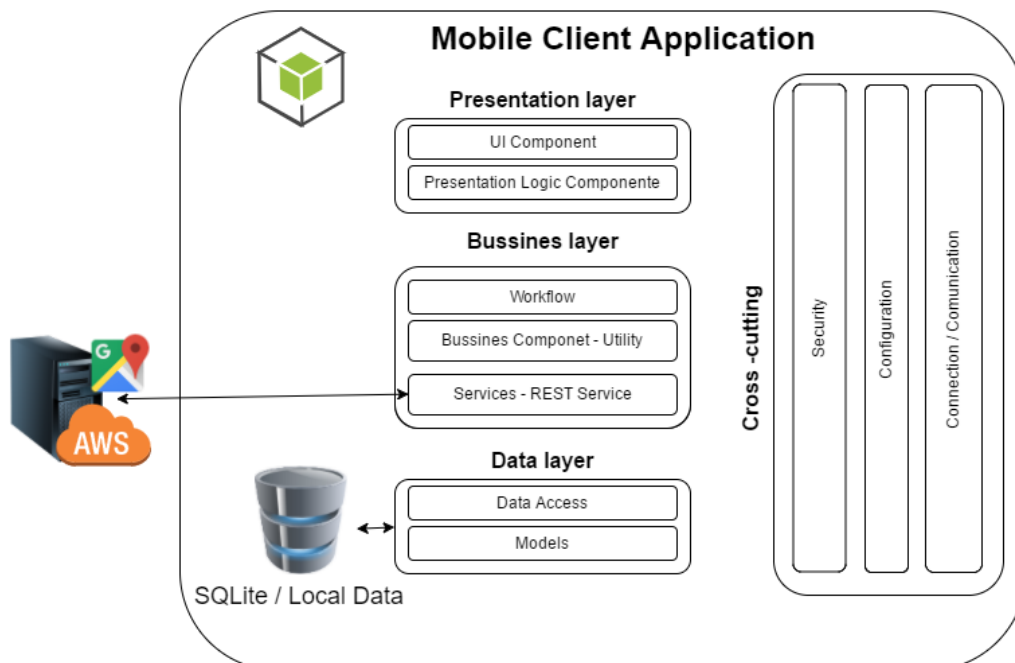


Figura 23. Arquitectura de la Aplicación Móvil. Elaboración propia.

Diseño de Interfaces para Aplicación Móvil

La aplicación móvil está compuesta por interfaces de ingreso a la aplicación, interfaz principal de la aplicación, interfaces secundarias e interfaces utilitarias. En las interfaces principal y secundarias de la aplicación se presentan los componentes de menú, diálogos de confirmación y diálogos de progresos.

Diseño de interfaz de ingreso a la aplicación

La interfaz está compuesta por una imagen de la aplicación y un formulario de ingreso al sistema.

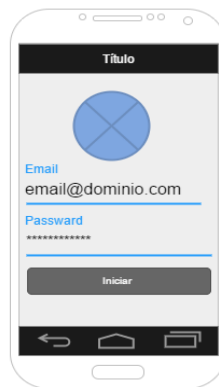


Figura 24. Interface de ingreso a la aplicación. Elaboración propia.

Diseño de interfaz principal de la aplicación

Esta interfaz es la pantalla principal de la aplicación en la cual se muestra un mapa con la ubicación actual del usuario y el de la mascota en el caso de que haya sido buscada. Esta interfaz también muestra un mensaje de inicio.



Figura 25. Interfaz de principal de la aplicación. Elaboración propia.

Diseño de interfaces complementarias de la aplicación.

Las interfaces complementarias o de funcionalidades específicas en la cual se define las administraciones de la aplicación.

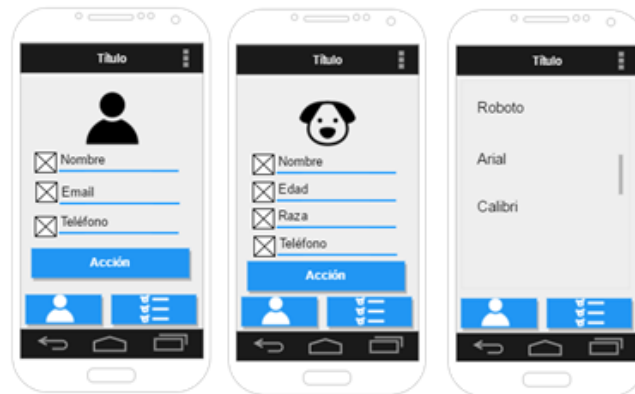


Figura 26. Interfaces complementarias de la aplicación. Elaboración propia.

Diseño de interfaces de utilitarias de la aplicación

Estas interfaces muestra información de la aplicación o "splash screen" de la aplicación.

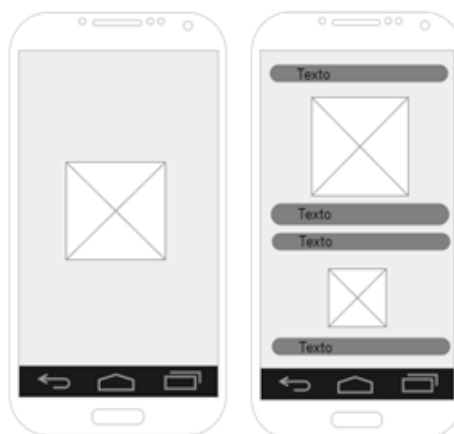


Figura 27. Interfaces utilitarias. Elaboración propia.

Diseño de componentes de la aplicación

Para mensajes de confirmación se ha definido el uso de Diálogos de alertas, para mensajes instantáneos se definió el uso de Toast, para el uso de procesos en progreso se definió el uso de Alertas de progreso.



Figura 28. Componentes de la aplicación. Elaboración propia.

Diagrama de Secuencias

El diagrama de secuencia muestra las iteraciones entre los actores, componentes y subsistemas, mostrando en el tiempo el flujo de control de un participante a otro. En la siguiente figura de secuencia muestra de mejor manera el flujo de control de la solución de software.

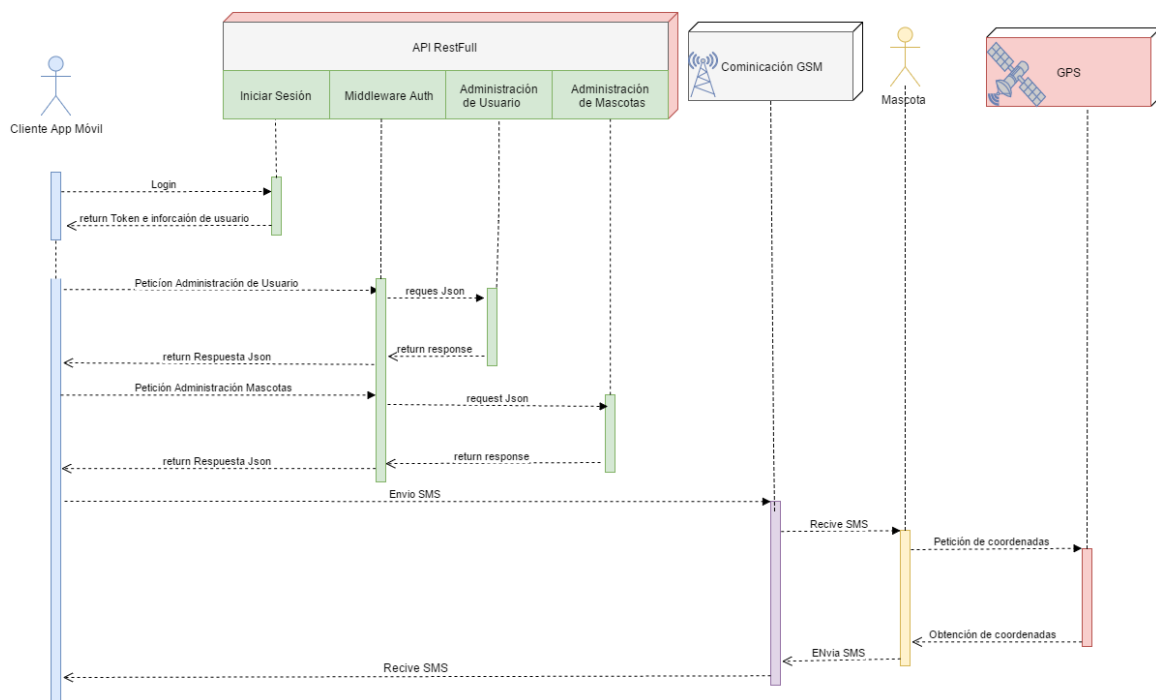


Figura 29. Diagrama de secuencia. Elaboración propia.

Gestión de Tareas

Para la gestión de tareas se realizará mediante la utilización de la herramienta *trelllo*, en la cuales se maneja los siguientes estados para las tareas: Hacer, en progreso, en revisión, hecho, cerrado.

Hacer: Aquí estarán las tareas o requerimientos que se necesitan desarrollar.

En progreso: Las tareas que se ubicarán aquí, serán las que se desarrollarán actualmente.

En revisión: En esta sección se ubicarán las tareas que entrarán a prueba de funcionalidad, si no pasan regresarán a la etapa de “hacer”.

Hecho: En esta etapa se situarán las tareas que pasaron la etapa de “en revisión”.

Cerrado: En esta etapa se encuentran todas las tareas finalizadas.

Guía para la utilización del dispositivo de GPS tracker

El dispositivo trabaja basado en las redes GSM/GPRS y satélites GPS. Además permite localizar y monitorear cualquier objetivo remoto mediante SMS. En las siguientes tablas se muestran las especificaciones, indicadores y funcionamiento del dispositivo.

Tabla 5. Especificaciones del Dispositivo GPS. Elaboración propia.

Contenido	Especificaciones
Dimensiones	70mm x 37mm x 20mm
Peso	44g
Redes	GSM/GPRS
Bandas	850/900/1800/1900 Mhz
Sensibilidad GPS	-159 dBm
Exactitud GPS	5m
Batería	Batería cambiable 3.7V 1000mAh Li-on
Tiempo de batería	300 horas

Tabla 6. Indicadores del dispositivo GPS. Elaboración propia.

Estado del indicador Led	Significado
VERDE - Sólido	No hay señal GPRS
VERDE - Intermitente	Comunicación Estable GPRS
ROJO – No Iluminado	Carga completa
ROJO – Sólido	Cargando
AZUL – Sólido	Buscando señal GPS
AZUL – Intermitente	Ubicación GPS exitosa

Tabla 7. Comando de funcionamiento del dispositivo GPS. Elaboración propia.

Instrucción	Comando	Respuesta
Reiniciar el dispositivo	Begin123456	Begin ok
Establecer número autorizado	Admin123456 phone number	Admin ok
Cancelar numero autorizado	Noadmin123456	Noadmin ok
SMS track	G123456#	Google link
Cambiar al modo sin GPRS	Nogprs123456	Nogprs ok
Cambiar a modo GPRS	Gprs123456	Grps ok
Chequear estado	Check123456	ID, IP, APN, GPS, GSM status
Establecer nueva APN	Apn123456 apn	Apn ok
Establecer username de APN	Apn123456 username	Apnuser ok
Establecer password de APN	Apnpasswd123456 password	Apnpasswd ok

3.9.3 Fase de Producción

En esta fase se genera el código de la aplicación móvil, para la planificación de tareas se implementa *trello* en las que se define las característica de la aplicación a desarrollar en forma de “Historias de usuarios” las cuales están en link⁷, el código se adjuntará en un dispositivo electrónico a este trabajo.

⁷ Tablero de tareas e historias de usuario para este proyecto: <https://trello.com/b/cRo83yoR/app-pettracker>

En el desarrollo de solución de la API RESTful se implementó el módulo de seguridad JWT (*Json Web Token* ver **Anexo F**) que utiliza el estándar RFC 7519, además para la definición de los log y errores se utilizó el estándar RFC 5424 y para la utilización de los métodos de comunicación HTTP se aplicó el estándar RFC 2616 los cuales ya vienen integrados para su utilización en los proyectos de Laravel.

Para el desarrollo de la solución móvil se utilizó el estándar de diseño visual denominado “*Material Design*” propuesto por Google, el enlace hacia la página oficial se encuentra en el **Anexo F**.

El diseño de la base de datos se encuentra en el **Anexo C**.

Para la integración de la aplicación móvil y el API, se define los recursos expuestos en el archivo SPECS adjunto a este proyecto **Anexo G**.

En la construcción de la solución también se creó como parte de la solución un sistema web para la gestión de datos por parte de los usuarios finales ver **Anexo H**, esta aplicación web fue construido mediante el patrón de arquitectura de software (MVC “Modelo-Vista-Controlador”). Utiliza la misma arquitectura que el servidor web ver **Figura 22**.

3.9.4 Fase de estabilización

En esta fase se define las mejoras de calidad del producto implementadas. También se realiza la documentación del proyecto basado en el patrón *Wrap-Up* (Resumen), para la Interfaz de usuario, arquitectura de software y diseño de la solución.

Requerimiento de hardware y software de móviles

Tabla 8. Requerimiento de Dispositivo Móvil. Elaboración propia.

Dispositivo Móvil	
Hardware	Servicio de Ubicación, Memoria RAM 4 GB.
Software	Versión Android mayor o igual a 6.0

Tabla 9. Requerimiento de Servidores para API RESTful. Elaboración propia.

Servidores de la API RESTful	
EC2 Instancia t2.micro de AWS	Sistema Operativo: Centos PHP 5.6 Servidor Web :Apache 2.4.6 Espacio en disco :1 GB Memoria RAM mínimo 2 GB
RDS Instancias MySQL de AWS	MySql 5.6.27 Espacio en disco: 2 GB

Se adjunta la documentación de la arquitectura del software, base de datos, script, historias de usuario, guía de uso del dispositivo GPS de la mascota y códigos fuente desarrollados a este trabajo.

3.9.5 Fase de pruebas y arreglos del sistema

Para la validación y verificación de la solución se procedió a realizar pruebas de desempeño y estrés para los servidores donde se encuentra alojada la API *RESTful* para la cual se utilizara la herramienta *Apache Bench* y pruebas de funcionalidad en la aplicación móvil.

Pruebas de Carga: Proceso que se le impone a un sistema basado en una cantidad predefinida de peticiones o procedimientos con la finalidad de determinar su comportamiento esperado para dicha situación.

Prueba de Estrés: Determina el comportamiento del sistema bajo un nivel de exigencia mayor al capaz de manejar.

Para las pruebas se realizaron 1000 peticiones generadas desde un usuario concurrente las cuales se incrementaron hasta generar bajo rendimiento del servidor, en la cual al final se llegó hasta tener 100 usuarios concurrentes generando cada uno 1000 peticiones al servidor al mismo tiempo, lo que se interpreta como 100000 peticiones lanzadas al servidor.

Como resultados importantes se obtuvo que la media de request por segundo fue de 3.36, la media para cumplir una petición es de 29732.062 milisegundos, la media de la velocidad de transferencia es de 297.631 milisegundos, el tiempo por estado de petición es de 9.32 Kbites

por segundos, request fallidos es igual 0 y la media de tiempo empleado para conectarse al servidor fue 0.306 segundos; los cuales se puede apreciar en la figura 29.

La figura 30 demuestra el comportamiento del servidor antes la prueba de stress en el que se puede evidenciar que dado el número de peticiones al servidor este generó una cola de botella para que el tiempo de respuesta no aumente más allá de lo permitido cómo aceptable, además responde a todas las peticiones generadas.

```
C:\xampp\apache\bin>ab -n 1000 -c 100 -g test.tsv http://ec2-52-67-34-88.sa-east-1.compute.amazonaws.com/
This is ApacheBench, Version 2.3 <Revision: 1706008>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ec2-52-67-34-88.sa-east-1.compute.amazonaws.com (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software:      Apache/2.4.18
Server Hostname:      ec2-52-67-34-88.sa-east-1.compute.amazonaws.com
Server Port:          80

Document Path:        /
Document Length:      2181 bytes

Concurrency Level:    100
Time taken for tests:  297.631 seconds
Complete requests:    1000
Failed requests:       0
Total transferred:    2839000 bytes
HTML transferred:     2181000 bytes
Requests per second:  3.36 [#/sec] (mean)
Time per request:     29763.062 [ms] (mean)
Time per request:     297.631 [ms] (mean, across all concurrent requests)
Transfer rate:        9.32 [Kbytes/sec] received

Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:    209    297 55.1    306   1230
Processing:  276 27971 5235.2 29325 30838
Waiting:    274 14945 8682.6 14950 30616
Total:      583 28267 5235.5 29598 31143

Percentage of the requests served within a certain time (ms)
 50%  29598
 60%  29982
 75%  30200
 80%  30310
 90%  30713
 95%  30823
 98%  30924
 99%  31101
100% 31143 (longest request)
```

Figura 30. Resultado del test de carga y stress del servidor. Elaboración propia.

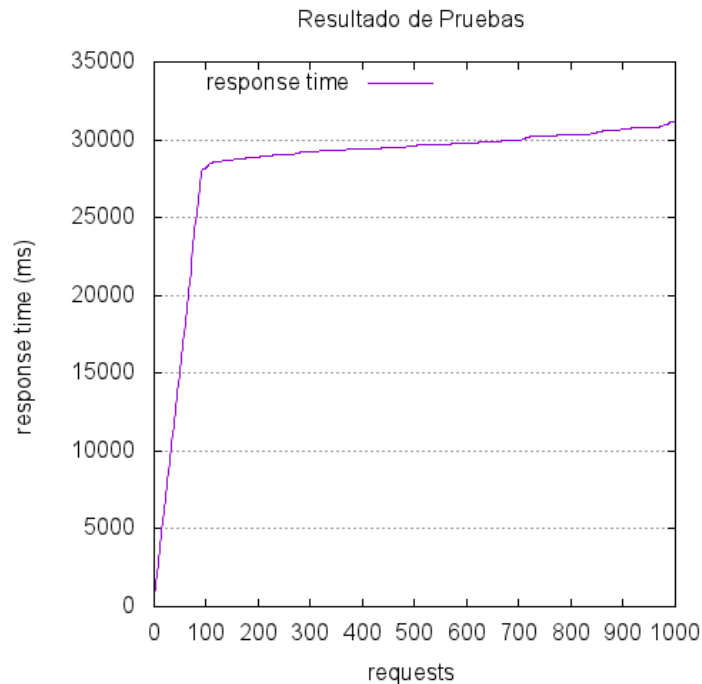


Figura 31. Diagrama de comportamiento del servidor. Elaboración propia.

Pruebas de funcionalidad de la aplicación móvil: En las pruebas de funcionamiento de la aplicación se pudieron evidenciar los siguientes aspectos:

- 1) La comunicación entre el dispositivo de la mascota y la aplicación móvil depende en gran medida de la intensidad de la señal GSM/GPRS.
- 2) Para la comunicación entre los dispositivos es imprescindible tener tiempo aire (crédito) ya que nos permite el uso de las redes de comunicación de las operadoras telefónicas para el envío y recepción de los SMS, tanto desde el dispositivo dónde se encuentra instalada la aplicación como desde el dispositivo GPS de la mascota; además debe estar activo el dispositivo GPS de la mascota.
- 3) El dispositivo GPS sólo puede tener las coordenadas en lugares abierto, sino se encuentra las coordenadas este devuelve las últimas coordenadas obtenidas, además de información adicional como fecha y hora y porcentaje de batería del dispositivo GPS.

Considerando los puntos antes mencionados las pruebas de funcionalidad de iniciar sesión, cerrar sesión, mostrar mapa, mostrar usuario, actualizar datos de usuario, mostrar lista de

mascotas, mostrar mascotas, mostrar información de la aplicación y buscar mascotas fueron satisfactorias.

El resultado final de la aplicación móvil es un producto funcional, el cual cumple con los requerimientos especificados; el nombre final de la aplicación móvil es “*FindBobby*”, los íconos y logos se realizaron en función del nombre, el **anexo D** muestran las pantalla finales de la aplicación.

El **Anexo E** muestra la liberación de aplicativo final, en él se muestra gráficas de las consolas de las instancias EC2 y RDS de Amazon donde corre la API RESTful y la base de datos, también muestra la aplicación móvil en la tienda de Google Play.

DISCUSIÓN

En la exploración de datos vemos como se sigue una misma tendencia nacional así como también en el marco internacional para satisfacer las necesidades que surgen hoy en día en el desarrollo de software y poder conseguir el resultado esperado en un entorno con requisitos muy volátiles.

También los datos nos muestran como la metodología predilecta de los gestores de proyecto y desarrolladores de software es SCRUM la cual es como tal una metodología de gestión de proyecto de software universal a diferencia de su semejante Mobile-D, el cual es una metodología de desarrollo de software enfocado a las aplicaciones móviles. La metodología Mobile-D a diferencia de las metodologías mencionadas en la comparación (ver tabla 3) se apegan más a las características que se identificaron para la gestión de proyectos de aplicaciones móviles y las cuales la hacen “ganadora” sobre las otras metodologías mencionadas en este trabajo (ver sección 3.8) tales como (ver tabla 3):

- ❖ Iteraciones con tiempos cortos.
- ❖ Identificación de usuarios finales.
- ❖ Minimización de impacto dada las limitaciones inherentes y de evolución de los dispositivos móviles(ver sección 1.3.1)
- ❖ Documentación.
- ❖ Valores y principios orientados a la elaboración de aplicaciones App.

Esto evidencia la importancia de tener una línea arquitectónica del proyecto bien establecida para la parte de back-end de la aplicación para un funcionamiento eficiente y eficaz, así como la parte del front-end ya que de esto depende que el usuario interactúe y le sea de fácil uso lo implica que la aplicación se aceptada por el usuario.

Un punto muy importante evidenciado en este trabajo es el ausentismo de estrategias de diseño de producto en las metodologías propuestas, ya que la gestión de proyectos móviles no se debe limitar a la pura escritura de código y su respectivo funcionamiento, sino también la viabilidad del proyecto, implementación de estrategias de mercado, etc; para poder sobrevivir en este mundo tan cambiante y competitivo.

Por otra parte no se siguió de forma estricta la metodología Mobile-D para la construcción de esta aplicación móvil dadas la limitaciones obtenidas, como fue el caso de número de participantes en el desarrollo de este proyecto el cual fue “uno”, pero se pudo conseguir el resultado esperado dadas las características de las metodologías ágiles de ser adaptivas y más aún en proyectos no tan grandes como son las aplicaciones móviles.

CONCLUSIONES

Las características tan distintivas del desarrollo de aplicaciones móviles (MAD) implican la necesidad del uso de metodologías ágiles debida a las altas incertidumbres en la construcción de software y las limitaciones tecnológicas de los móviles mencionadas en la sección 1.3.1, para lograr minimizar el impacto de estas restricciones.

Se analizó las metodologías ágiles (ver sección 3.8) en función de las necesidades para el desarrollo de aplicaciones móviles sea idónea y pertinente y en el cual se pudo evidenciar las características que se debe tener cuenta para que la implementación de estas sea eficaz y exitosa.

Como resultado del análisis realizado se implementó la metodología Mobile-D en la construcción de la aplicación móvil (ver sección 3.9), ya que esta se proyectó con mejores resultados en el estudio comparativo que se desarrolló en la presente investigación (ver tabla 3 y 4), demostrando la utilidad que esta tiene en la gestión de proyectos para aplicaciones móviles.

Es fundamental la importancia que se debe hacer a la interfaz de usuario (UI) y experiencia de usuario (UX), ya que estas características deben ser efectivas en las aplicaciones móviles para lograr el impacto que se desea en los usuarios finales. En el MAD la entrega temprana del producto es indispensable ya que uno de los puntos importantes es posesionarse en el mercado.

Mobile-D está orientada al desarrollo de aplicaciones móviles ya que hace énfasis en la línea arquitectónica, experiencia de usuario, orientación de la construcción pensada para el usuario final, iteraciones cortas pensadas para la construcción de funciones específicas; además de implementar etapas de inicialización y estabilización las cuales se las pueden crear o establecer en cualquier metodología ágil, pero que en esta metodología ya vienen integrada, estas características que proporciona la metodología MOBILE-D hacen que resulte ganadora entre las metodologías propuestas en este estudio (ver tabla 4).

Se desarrolló la aplicación para la búsqueda y localización de mascotas perdidas con resultados positivos mediante los procesos de gestión de la metodología antes mencionadas.

La utilización de una metodología de desarrollo de software depende de muchos casos de las características, requisitos y limitaciones del proyecto, por lo que la metodología se convierte en una referencia para la construcción del proyecto y no en una receta estricta a seguir, por lo que al final del día la experiencia obtenida en la gestión y desarrollo de proyectos de aplicaciones móviles determinará cuál será la mejor alternativa a implementar (fases, etapas, principios, valores etc. de una metodología, partes de esta o combinación de distintas metodologías).

Por otra parte existe la dificultad de que no hay límites claros de implementación de las metodologías ágiles, por lo que existe una complejidad al querer compararlas y poder emitir algún criterio sobre su uso.

RECOMENDACIONES

Para el uso correcto de metodologías ágiles en el desarrollo de aplicaciones móviles es fundamental un gestor de proyecto experimentado, así como personal con experiencia en el manejo de tecnologías móviles ya que este proveerá la mejor implantación disponible para el avance del proyecto.

Se debe adecuar las características necesarias para el desarrollo de aplicaciones móviles en las distintas metodologías existentes, ya que no son estrictas en su uso, convirtiéndola en una metodología de requisitos que demanden el cumplimiento de las necesidades de los dispositivos móviles.

En la construcción de aplicaciones móviles es recomendable tener de ante mano una visión completa de los compones tecnológicos que integran la solución de software ya que esto reduce el riesgo de generación de problemas en el desarrollo o avance del proyecto.

Que las futuras investigaciones sobre este tema, implementen el desarrollo de aplicaciones móviles con las diferentes metodologías propuestas y así poder obtener una visión más clara sobre el comportamiento que pueden tener cada una de estas metodologías con respecto al MAD.

BIBLIOGRAFÍA

1. ABRAHAMSSON, P. (2005). Keynote: Mobile software development -the business opportunity of today, 20–23.
2. ABRAHAMSSON, P. (2005). Mobile software development the business opportunity of today. Proceedings of the International Conference on Software Development (pp. 22–33). Reykjavik.
3. ABRAHAMSSON, P., HANHINEVA, A., HULKKO, H., IHME, J., KORKALA, M. (2004). Mobile-D: an agile approach for mobile application development. (pp. 174–175). New York. Recuperado de <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWVpbnxhc2VyZ3JwfGd4OjUwN2Y1NmI2M2QxZGNIYmU>
4. ABRAHAMSSON, P., VVT. (2010). MOBILE-D. Recuperado de <http://agile.vtt.fi/mobiled.html>
5. AGILE ALLIANCE. (2001). Manifesto for Agile Software Development. Recuperado de <http://agilemanifesto.org/>
6. AMAYA BALAGUERA, Y. D. (2013). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. *Revista de Tecnología | Journal Technology*, 12 número, 111–124. Recuperado de http://www.ae-ic.org/tarragona2012/contents/comunicacions_cd/ok/171.pdf
7. ANDERSON, D. J., REINERTSEN, D. G. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. (Blue Hole Press, Ed.).
8. BERMEJO, M. (n.d.). El Kanban. Recuperado de [https://www.exabyteinformatica.com/uoc/Audiovisual/Produccion_multimedia/Produccion_multimedia_\(Modulo_4\).pdf](https://www.exabyteinformatica.com/uoc/Audiovisual/Produccion_multimedia/Produccion_multimedia_(Modulo_4).pdf)
9. BLANCO, P., CAMARERO, J., FUMERO, A., WERTERSKI, A., RODRÍGUEZ, P. (2009). *Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone*. Madrid. Recuperado de http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf

10. BROULLÓN, A. (2013). Kanban digital: gestionando trabajo en equipo. Recuperado de <https://mangelesbroullon.wordpress.com/2013/06/24/kanban-digital-gestionando-trabajo-en-equipo/>
11. CALDERÓN, A., DÁMARIS, S., V. R. J, Carlos. (2007). *Metodologías Ágiles*. Trujillo. Recuperado de [https://uvirtual.unet.edu.ve/pluginfile.php/268695/mod_resource/content/1/Metodologias Ágiles.pdf](https://uvirtual.unet.edu.ve/pluginfile.php/268695/mod_resource/content/1/Metodologias%20Ágiles.pdf)
12. CEREZO, J. M. (2010). Smartphone Toda la información al alcance de tu mano. *Telos* 82, 97–100.
13. CORTÉS CORTÉS, M., IGLESIAS LEÓN, M. E. (2004). *Generalidades sobre Metodología de la Investigación* (Primera). Ciudad del Carmen.
14. COSTA SÁNCHEZ, C., DÍAZ-GONZÁLEZ, M. J., VIDELA RODRÍGUEZ, J. J. (2012). La investigación sobre dispositivos móviles desde las Ciencias de la Comunicación: Aportaciones realizadas y retos futuros. In *Contextos móviles y fenómenos emergentes*. Tarragona.
15. DEEMER, P., BENEFIELD, G., BAS, C. L., VERSIÓN, V. (2009). *Información básica de scrum (The Scrum Primer)*. San Francisco. Recuperado de http://www.goodagile.com/scrupprimer/scrupprimer_es.pdf
16. DONVAN WELL, J. (2000). Extreme Programming: A gentle introduction. Recuperado de <http://www.extremeprogramming.org/index.html>
17. ECHEVERRY TOBÓN, L., DELGADO CARMONA, L. (2007). *Caso práctico de la metodología ágil XP al desarrollo de software*. Universidad Tecnológica de Pereira. Recuperado de <http://recursosbiblioteca.utp.edu.co/dspace/bitstream/11059/794/1/0053E18cp.pdf>
18. ERICSSON. (2015). ON THE PULSE OF THE NETWORKED SOCIETY Ericsson Mobility Report.
19. FERNÁNDEZ GONZÁLEZ, J. (2013). *Introducción a las metodologías ágiles Otras formas de analizar y desarrollar* (No. PID_00184468). Recuperado de [https://www.exabyteinformatica.com/uoc/Informatica/Tecnicas_avanzadas_de_ingenieria_de_software/Tecnicas_avanzadas_de_ingenieria_de_software_\(Modulo_3\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnicas_avanzadas_de_ingenieria_de_software/Tecnicas_avanzadas_de_ingenieria_de_software_(Modulo_3).pdf)
20. FLORA, H. K., CHANDE, S. V. (2013). A Review and Analysis on Mobile Application Development Processes using Agile Methodologies. *International*

- Journal of Research in Computer Science*, 3(4), 2249–8265.
<http://doi.org/10.7815/>
21. FLORA, H. K., WANG, X., CHANDE, S. V. (2014). An Investigation into Mobile Application Development Processes: Challenges and Best Practices. *I.J. Modern Education and Computer Science Modern Education and Computer Science*, 6(6), 1–9. <http://doi.org/10.5815/ijmecs.2014.06.01>
 22. GIMSON, L. (2015). *Desarrollo basado en conocimiento siguiendo prácticas ágiles*. UNIVERSIDAD NACIONAL DE LA PLATA. Recuperado de http://sedici.unlp.edu.ar/bitstream/handle/10915/50431/Documento_completo___.pdf-PDFA.pdf?sequence=3
 23. HAYES, I. S. (2003). *Just enough wireless computing*. Prentice Hall.
 24. JOSKOWICZ, J. (2008). *Reglas y Prácticas en eXtreme Programming*. Vigo: Universidad de Vigo. Recuperado de <http://iie.fing.edu.uy/~josej/docs/XP - Jose Joskowicz.pdf>
 25. KNIBERG, H., SKARIN, M. (2010). *Kanban y Scrum – Obteniendo lo mejor de ambos*. Madrid: Projectails. Recuperado de http://www.projectalis.com/documentos/KanbanVsScrum_Castellano_FINAL-printed.pdf
 26. LETELIER, P., PENADÉS, M. C. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). www.cyta.com.ar/ta0502/v5n2a1.htm. Recuperado de <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
 27. MARIÑO, S. I., GODOY, M. V., ALFONZO, P. L. (2014). Propuestas y revisión de metodologías de la Ingeniería del Software. In *XVI Workshop de Investigadores en Ciencias de la Computación* (pp. 377–381). Corrientes: Red de Universidades con Carreras en Informática (RedUNCI). Recuperado de http://sedici.unlp.edu.ar/bitstream/handle/10915/41409/Documento_completo.pdf?sequence=1
 28. NAVARRO C., A., FERNÁNDEZ M., J. D., MORALES V., J. (2013). *30 Revisión de metodologías ágiles para el desarrollo de software*. Cali. Recuperado de <http://es.scribd.com/doc/242134315/Dialnet-RevisionDeMetodologiasAgilesParaElDesarrolloDeSoft-4752083#scribd>
 29. OIVER, A., PÉREZ, A. (2011). Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM. *Revista Inventum*, 0(10).
 30. PERALTA, A. (2003). *Metodología SCRUM*. Montevideo. Recuperado de

- <http://fi.ort.edu.uy/innovaportal/file/2021/1/scrum.pdf>
31. PONCE, L. (1996). Metodología de la investigación científica del derecho (pp. 61–73). Revista de la Facultad de Derecho, Universidad Nacional Autónoma de México. Recuperado de <http://www.juridicas.unam.mx/publica/librev/rev/facdermx/cont/205/dtr/dtr4.pdf>
 32. POOLE, D. B. (2009). *Do It Yourself Agile* (Segunda).
 33. RAYA, R. (2014). ¿Qué son las metodologías ágiles? Recuperado de <http://blog.leanmonitor.com/es/que-son-las-metodologias-agiles/>
 34. REVELO, J. (2014). ¿Cómo Empezar A Desarrollar Aplicaciones Android? Retrieved January 9, 2017, from <http://www.hermosaprogramacion.com/2014/08/android-programar-app-como/>
 35. REVISTA LIDERES. (n.d.). Los servicios de las aplicaciones se expanden en Ecuador. *Revista Lideres*. Recuperado de <http://www.revistalideres.ec/lideres/servicios-aplicaciones-expanden-ecuador.html>
 36. SCHWABER, K., Sutherland, J. (2013). La Guía de Scrum La Guía Definitiva de Scrum: Las Reglas del Juego. Recuperado de <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
 37. SIFUENTES, A. M., TARELO, B. A. G., SALDAÑA, J. F. E., DEMOSS, M. V. G., MADRAZO, B. J. M. (2016). Prototipo de herramienta de desarrollo multiplataforma para dispositivos móviles. *CULCyT*, 0(57).
 38. SPATARU, A. C. (2010). Agile Development Methods for Mobile Applications. Recuperado de <https://www.inf.ed.ac.uk/publications/thesis/online/IM100767.pdf>
 39. STATCOUNTER GLOBAL STATS. (2015). StatCounter Global Stats. Recuperado de <http://gs.statcounter.com/>
 40. VERA, M. (2014). Intelligence to Business: Los beneficios de Implementar la Metodología Ágil. Recuperado de <http://www.i2btech.com/blog-i2b/tech-deployment/los-beneficios-de-implementar-la-metodologia-agil/>
 41. VERSION ONE. (2009). *State Of Agile Survey. Annals of Physics* (Vol. 54). Recuperado de <http://stateofagile.versionone.com/>
 42. WEINBERG, G. (1992). *Quality Software Management: Systems Thinking*. New York: DORSET HOUSE PUBLISHING.

ANEXO

ANEXO A

Encuesta de metodologías ágiles

Encuesta realizada a 38 desarrolladores de software de distintas empresas (Cobiscorp, Sloncorp, Kruger Corporation, Rutatec, BigData C.A., TATA Consultancy Services, LeadSolutions Cia Ltda.) de la ciudad de Quito, Ecuador. Elaborado por Wilmer Zambrano García.

Datos sobre el uso de las metodologías ágiles

***Obligatorio**

1. Qué metodología se usa más en su trabajo? *

Marca solo un óvalo.

- ☐ Scrum
- ☐ XP
- ☐ Kanban
- ☐ Metodologías ágiles personalizadas
- ☐ Otro: _____

2. Cuáles fueron las mejoras que se evidenciaron al implementar metodologías ágiles?. *

Selecciona todos los que correspondan.

- ☐ Habilidad para gestionar cambios prioritarios
- ☐ Equipos productivos
- ☐ Visibilidad del proyecto
- ☐ Una mayor previsibilidad de entrega
- ☐ Mayor calidad de software
- ☐ Reducción del riesgo del proyecto
- ☐ Otro: _____

3. Causas que impiden adoptar la metodología ágil? *

Selecciona todos los que correspondan.

- ☐ No existe personal con la suficiente experiencia
- ☐ Dificultad sobre el cambio de la cultura organizativa
- ☐ Resistencia al cambio
- ☐ Problemas en la gestión por falta de planificación
- ☐ Pérdida de control en la gestión
- ☐ Problemas de comunicación
- ☐ Otro:

4. Cómo mide el éxito de haber usado una metodología ágil? *

Selecciona todos los que correspondan.

- ☐ Tiempo de entrega
- ☐ Satisfacción del cliente/usuario
- ☐ Valor del negocio
- ☐ Visibilidad del proyecto
- ☐ Productividad
- ☐ Otro:

5. Técnicas de metodologías ágiles que más utiliza? *

Selecciona todos los que correspondan.

- ☐ Standup diario
- ☐ Planificación de las Iteraciones
- ☐ Backlog priorizados
- ☐ Iteraciones cortas
- ☐ Test unitarios
- ☐ Programación en par
- ☐ Refactorización
- ☐ TDD
- ☐ Integración continua
- ☐ Normas de codificación
- ☐ Otro:

ANEXO B

Entrevista a desarrolladores y gestores de proyectos

- Entrevista número uno.

Nombre: Fabio Puente Burbano.

Experiencia:

- Ingeniero en Sistemas Y Computación - Pontificia Universidad Católica Del Ecuador
- Master en Redes Y Comunicaciones - Pontificia Universidad Católica Del Ecuador
- Presidente del Colegio de Ingenieros en Sistemas Y Computación de Pichincha- Ecuador periodo (2006 – 2010), (2016 – Actual)
- Director del Parque Tecnológico Machangarasoft
- Quince años de experiencia como profesor de la Pontificia Universidad Católica del Ecuador PUCE, Universidad De Las Fuerzas Armadas ESPE, Universidad Tecnológica Del Ecuador UTE.

1. ¿Implementa una metodología ágil en la construcción de aplicaciones móviles?

Fabio Puente refirió que normalmente utilizan una metodología cuando la dimensión o el tamaño del desarrollo de la aplicación la requieren, en el caso que sea necesario se utiliza SCRUM, dado que en las aplicaciones con requerimientos puntuales no es necesario una metodología, como buena práctica utiliza la programación en parejas.

El gestor de proyectos identifica los requerimientos y utiliza una mezcla de buenas prácticas de varias metodologías rescatando y adoptando a las necesidades.

2. ¿Según la experiencia que usted ha tenido, cuál es la metodología que se acopla de mejor manera al desarrollo de aplicaciones móviles?

Él define una aplicación mixta de las metodologías, no recomienda una metodología formal, ya que la naturaleza y las variables que determinan las necesidades del proyecto no son completamente estables.

3. ¿Por su experiencia, qué fases o valores de ciertas metodologías que han utilizado le parece que se acoplan de mejor manera a lo que es una aplicación móvil?

Fabio Puente citó que la determinación de los requerimientos no debe ser fallida, deben estar bien definidos los requerimientos y determinar el alcance de la aplicación, él mencionó que las fases y valores a aplicar dependerán de la aplicación a evaluar.

4. ¿Dadas las restricciones y limitaciones inherentes de hardware y software en el desarrollo de aplicaciones móviles qué metodología se ajusta de mejor manera a la resolución de estos problemas?

Fabio Puente mencionó que las metodologías no están destinadas en para la utilización exclusiva para el desarrollo de aplicaciones móviles, las consideraciones de restricciones y limitaciones de hardware y software son de carácter técnico el cual no considera parte de una metodología; el también citó restricciones y limitaciones directamente vinculadas con el producto final como es el caso del nombre de la aplicación móvil en el cual puede ser negada por alguna de plataformas de aplicaciones móviles. El considera que este tipo de consideraciones técnicas pueden ser adaptadas en la mayoría de las metodologías ágiles.

- Entrevista número dos.

Nombres: Brayan Alexander Burgos Suarez

Experiencia:

- Ingeniero en Sistemas de la Pontificia Universidad Católica Del Ecuador
- Cuatro años de experiencia en el desarrollo web y desarrollo móvil.
- Gerente de proyectos de la empresa Sloncorp.

1. ¿Implementa una metodología ágil en la construcción de aplicaciones móviles?

Él menciona que pone en práctica la utilización de metodologías ágiles en proyectos largos o cuando los requerimientos están no están totalmente bien definidos, él citó que generalmente utiliza las metodologías ágiles eXtreme Programming o SCRUM.

2. ¿Según la experiencia que usted ha tenido, cuál es la metodología que se acopla de mejor manera al desarrollo de aplicaciones móviles?

Brayan Burgos mencionó que entre las dos metodologías que él utiliza (eXtreme Programming - SCRUM) tienen pros y contras, citó que SCRUM es extremadamente útil cuando tiene trabajo conjunto con el cliente, cuando no está bien definido los requerimientos, y necesitas ir mejorando los requerimientos sobre la marcha, también mencionó que otro lado eXtreme Programming no está atado a sostener reuniones diarias o el involucramiento con el cliente a un alto nivel sino al potenciamiento de grupo de desarrollo, y para la elección de una metodología dependerá del carácter del proyecto.

3. ¿Por su experiencia, qué fases o valores de ciertas metodologías que han utilizado le parece que se acoplan de mejor manera a lo que es una aplicación móvil?

Brayan Burgos mencionó que en todos los desarrollos de software la parte más complicada es al inicio, es decir el levantamiento de requerimiento, definir exactamente qué es lo que quiere el cliente, dijo que hacer un buen levantamiento de requerimiento es primordial para que las siguientes fases, citó que si se hace un buen levantamiento de requerimientos las siguientes fases serán solventadas y más fácil implementarlas, también aludió que las fases como diseño, implementación, pruebas, etc. son totalmente realizables en el desarrollo de aplicaciones móviles.

4. ¿Dadas las restricciones y limitaciones inherentes de hardware y software en el desarrollo de aplicaciones móviles qué metodología se ajusta de mejor manera a la resolución de estos problemas?

Él citó que las principales diferencias entre una aplicaciones web y las aplicaciones móviles es que trabaje online u offline, el consumo de datos y que la memoria del teléfono es mucho más pequeña que la de una computadora, además que el desarrollo de aplicaciones móviles puedes tener tantos dispositivos el cual varían en hardware y software, así que metodologías donde se implementen de buena forma fases de pruebas para la verificación de funcionalidades.

- Entrevistas número tres.

Nombre: Pablo Miguel López Mera.

Experiencia:

- Ingeniería en sistemas de la Universidad Técnica de Ambato
- Desarrollador de Software desde 1998
- Master Administración de Negocios MBA.
- Experiencia como Líder de proyectos
- Experiencia como Arquitecto SOA.
- Experiencia como SCRUM MASTER.

1. ¿Implementa una metodología ágil en la construcción de aplicaciones móviles?

Pablo López menciona que más que una metodología para el desarrollo de aplicaciones móviles utiliza un marco de trabajo o framework de trabajo como SCRUM o SCRUMBAN (Él no considera a las antes mencionadas “metodologías ágiles” metodologías de desarrollo de software).

2. ¿Según la experiencia que usted ha tenido, cuál es la metodología que se acopla de mejor manera al desarrollo de aplicaciones móviles?

Él mencionó que dentro del marco de trabajo que utiliza las prácticas ágiles tal como TDD (Test-Driven Development) ayudan de gran manera el desarrollo de aplicaciones móviles, sin embargo manifiesta que pueden haber ciertos límites cuando se requiere hacer pruebas tanto pruebas de funcionalidad como pruebas sobre interface gráfica del dispositivo, él manifiesta que estas pruebas se pueden realizar pero son más difíciles de implementar, las cuales estarían atadas a prácticas ágiles más que a una metodología en sí.

3. ¿Por su experiencia, qué fases o valores de ciertas metodologías que han utilizado le parece que se acoplan de mejor manera a lo que es una aplicación móvil?

Pablo López considera que la fase donde se consignan los requerimientos es trascendental en la construcción de aplicaciones móviles.

4. ¿Dadas las restricciones y limitaciones inherentes de hardware y software en el desarrollo de aplicaciones móviles qué metodología se ajusta de mejor manera a la resolución de estos problemas?

Pablo López menciona que en definitiva la fases en el que defines qué es lo que necesitas construir y cómo debe funcionar es primordial, dentro del desarrollo ágil existe la posibilidad de acoplar o crear etapas que conceptualicen esta necesidad, él menciona a modo de ejemplo crear un sprint cero en caso de SCRUM en las que se construye en working-skeleton el cual es un producto que no hace nada pero en el que se puede detectar problemas tempranos.

- Entrevistas número cuatro.

Nombre: David Vladimir Castro Cruz.

Experiencia:

- Ingeniero Informático de la Universidad Central del Ecuador
- Tres años de experiencia en el desarrollo web y móvil.
- Jefe de proyectos en la empresa Rutatec

1. ¿Implementa una metodología ágil en la construcción de aplicaciones móviles?

El menciona que sí, y que hoy en día es de vital importancia usar metodologías ágiles en la construcción de software ya que ayudan a responder de manera oportuna a las necesidades de los clientes.

2. ¿Según la experiencia que usted ha tenido, cuál es la metodología que se acopla de mejor manera al desarrollo de aplicaciones móviles?

David Castro menciona que dependen del tipo de proyecto la elección de una metodología como tal, y más bien hace énfasis en la utilización de prácticas ágiles en todo el proceso de desarrollo de aplicaciones móviles.

3. ¿Por su experiencia, qué fases o valores de ciertas metodologías que han utilizado le parece que se acoplan de mejor manera a lo que es una aplicación móvil?

David Castro menciona que la toma de requerimientos y la implementación de tecnologías en la construcción de aplicaciones móviles es de vital importancia

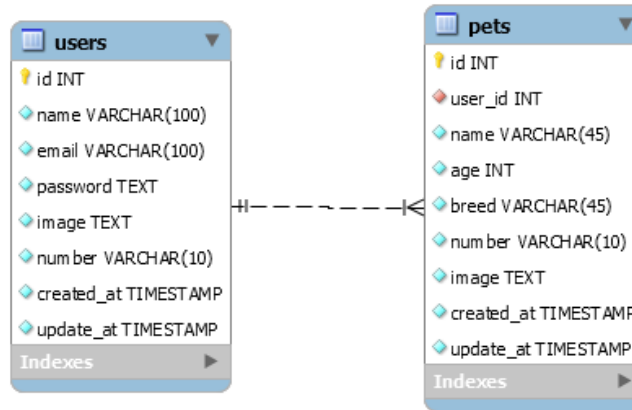
4. ¿Dadas las restricciones y limitaciones inherentes de hardware y software en el desarrollo de aplicaciones móviles qué metodología se ajusta de mejor manera a la resolución de estos problemas?

El mencionó que las metodologías de desarrollo ágil no tienen como objetivo resolver problemas de restricciones o limitaciones de hardware o de software, pero que sin duda se debe tener procesos ágiles que den una visión de las necesidades e inconvenientes tecnológicos en el desarrollo de aplicaciones móviles.

ANEXO C

Diagrama de base de datos y diccionario de datos

a) Diagrama de la base de datos



b) Diccionario de Datos

TABLA PETS

Tabla pets registran todas las mascotas que les pertenece a los usuario registrados y que hacen uso del servicio.

Nombre de la Columna	Tipo de Dato	PK	N N	U Q	BI N	U N	Z F	AI	Default	Comentario
id	INT	✓	✓					✓		ID identificador único de la tabla
user_id	INT		✓							FK de la tabla que designa a que usuario le pertenece la mascota
name	VARCHAR(45)	✓								Nombre de la mascota
age	INT	✓								Edad de la mascota
breed	VARCHAR(45)	✓								Raza de la mascota
number	VARCHAR(10)	✓								Número telefónico al cual está asociado al dispositivo GPS de la mascota
image	TEXT	✓								Foto de la mascota
created_at	TIMESTAMP	✓								Atributo para el log de la tabla
update_at	TIMESTAMP	✓								Atributo para el log de la tabla

TABLA USERS

Tabla users registra a los usuarios que tienen el servicio de la aplicación móvil.

Nombre de la Columna	Tipo de Dato	P K	NN	U Q	BIN	U N	ZF	A I	Defau It	Comentario
Id	INT	✓	✓					✓		ID identificador único de la tabla users.
name	VARCHAR(100)		✓							Atributo donde se registra los nombres de los usuarios.
email	VARCHAR(100)		✓	✓						Atributo donde se guarda el email del usuario este atributo sirve para iniciar sesión en la aplicación móvil.
password	TEXT		✓							Atributo para iniciar sesión en la aplicación móvil.
image	TEXT		✓							Atributo donde se guarda el path de la foto del usuario.
number	VARCHAR(10)		✓							Número telefónico del usuario que va a estar asociado a la aplicación móvil.
created_at	TIMESTAMP		✓							Atributo que sirve de los de la tabla.
update_at	TIMESTAMP		✓							Atributo que sirve de log de la tabla.

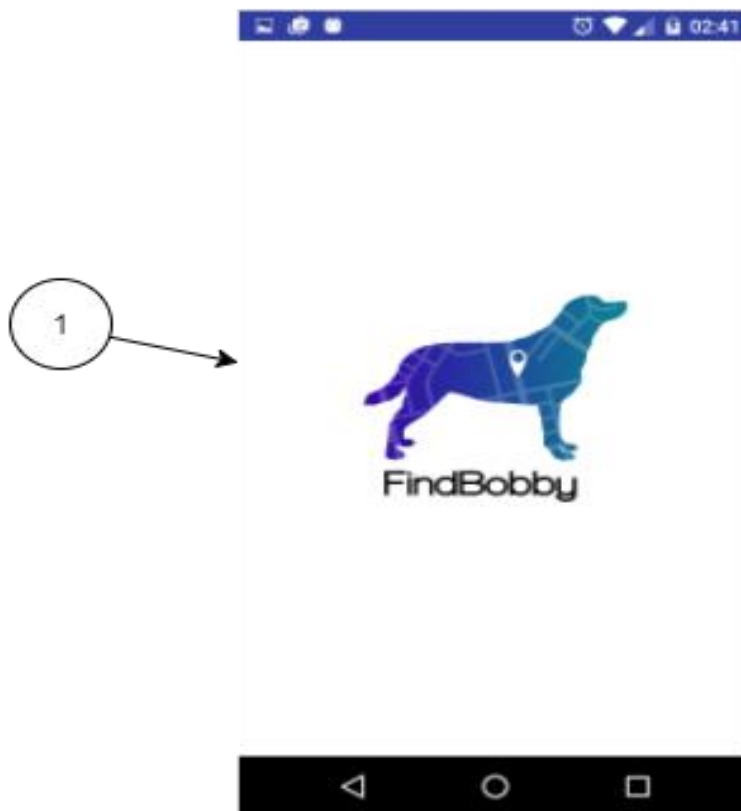
ANEXO D

Pantalla final de la aplicación

Descripción de las pantallas finales de la aplicación

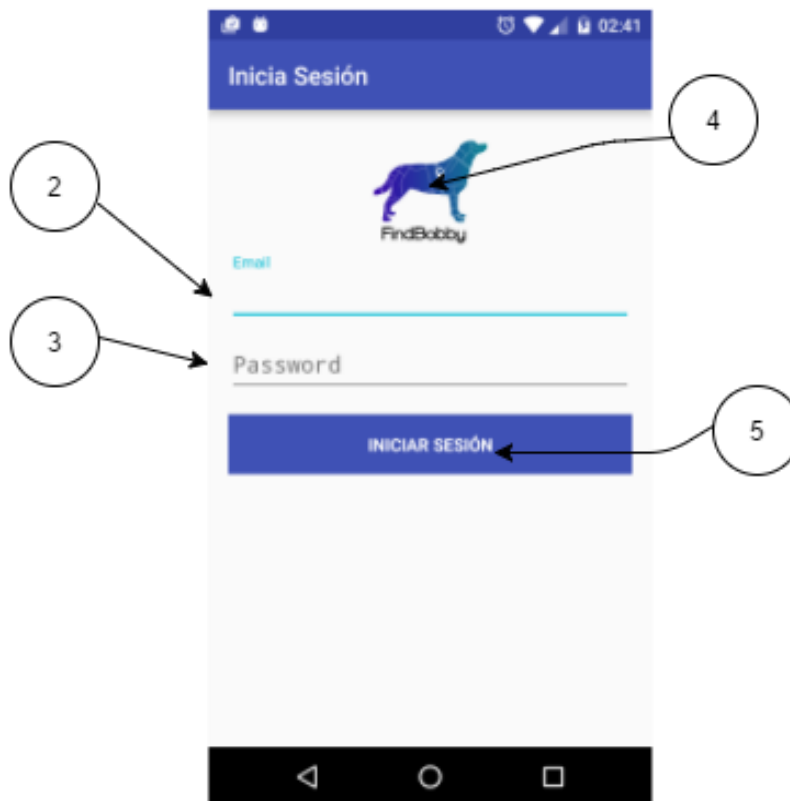
❖ **Pantalla Splash Screen:** Es la pantalla de inicio de la aplicación.

1) Logotipo de la aplicación



- ❖ **Pantalla de iniciar sesión:** Para poder utilizar la aplicación el usuario deberá iniciar sesión para poder obtener información de sus mascotas así como a las funcionalidades de la aplicación móvil.

- 2) Ingreso de email
- 3) Ingreso de password
- 4) Logotipo de aplicación
- 5) Botón de iniciar sesión



❖ **Ventana emergente de Bienvenida:** Esta ventana emergente busca interactuar con el usuario, generando la utilización de la aplicación.

6) Alerta que sugiere al usuario de realizar la función de la aplicación “Buscar las mascotas perdidas”



❖ **Pantalla principal:** Es la pantalla principal de la aplicación (HOME), desde esta pantalla podrá acceder a todas las opciones de la aplicación.

7) Opción menú “Sobre Nosotros”

8) Opción menú “Cerrar Sesión”

9) Botón de Ir a “Mi perfil”.

10) Botón de Ir a “Mis mascotas”

11) Pin que indica la posición de la mascota cuando se realizó su búsqueda.



- ❖ **Ventana emergente de “Cerrar sesión”:** Ventana emergente que permite cerrar la sesión del usuario.

12) Alerta de confirmación de “Cerrar sesión”, tienen como opciones “Sí” y “No”.



❖ **Pantalla “Sobre nosotros”:** Pantalla muestra información de la aplicación.

13) Logotipo de la aplicación.

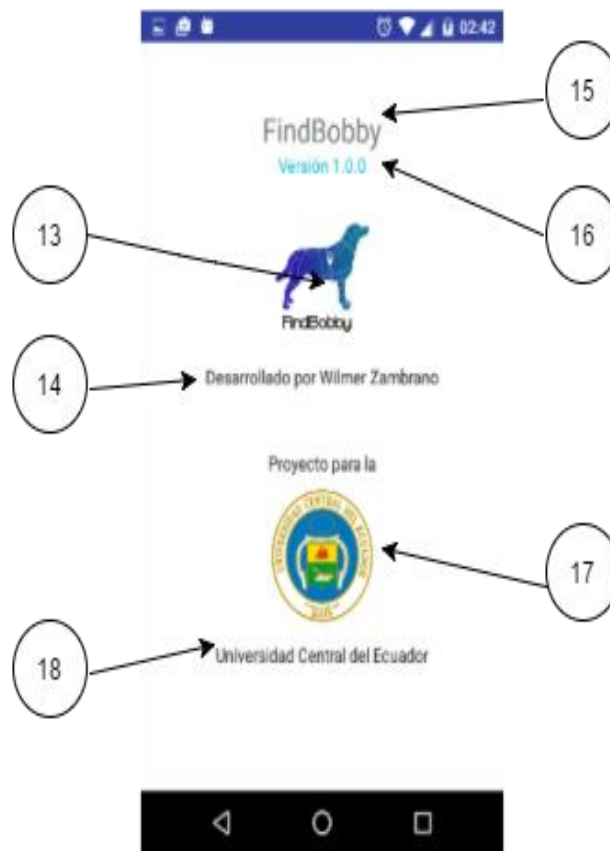
14) Por quién fue desarrollado la aplicación.

15) Nombre de la aplicación.

16) Versión de la aplicación.

17) Logotipo de la aplicación.

18) Nombre de la institución.



- ❖ **Pantalla de “Mis Mascotas”:** Muestra lista de mascotas registradas por el usuario.

19) Lista de las mascotas registradas para ese usuario.



❖ **Pantalla de “Mascota”:** Pantalla muestra datos de la mascota, desde esta pantalla podrá acceder a la búsqueda de la mascota seleccionada.

20) Nombre de la mascota

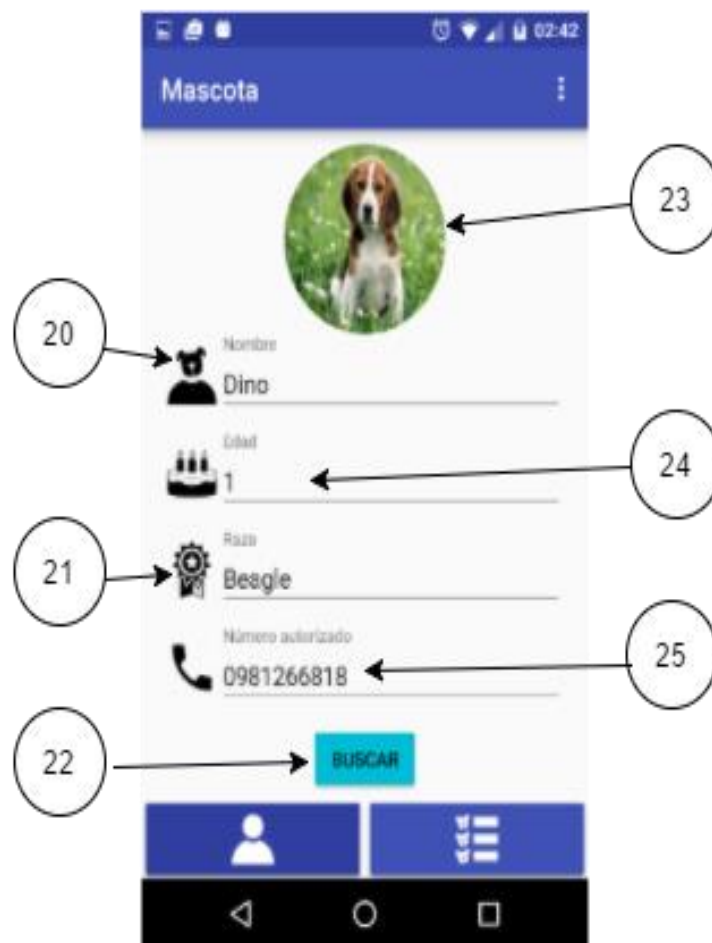
21) Raza de la mascota

22) Botón buscar mascota

23) Foto de la mascota

24) Edad de la mascota

25) Número telefónico asociado al dispositivo GPS de la mascota



❖ **Pantalla de “Mi Perfil”:** Pantalla muestra información del usuario. Esta pantalla tiene la opción de editar el nombre del usuario.

26) Foto del usuario de la aplicación.

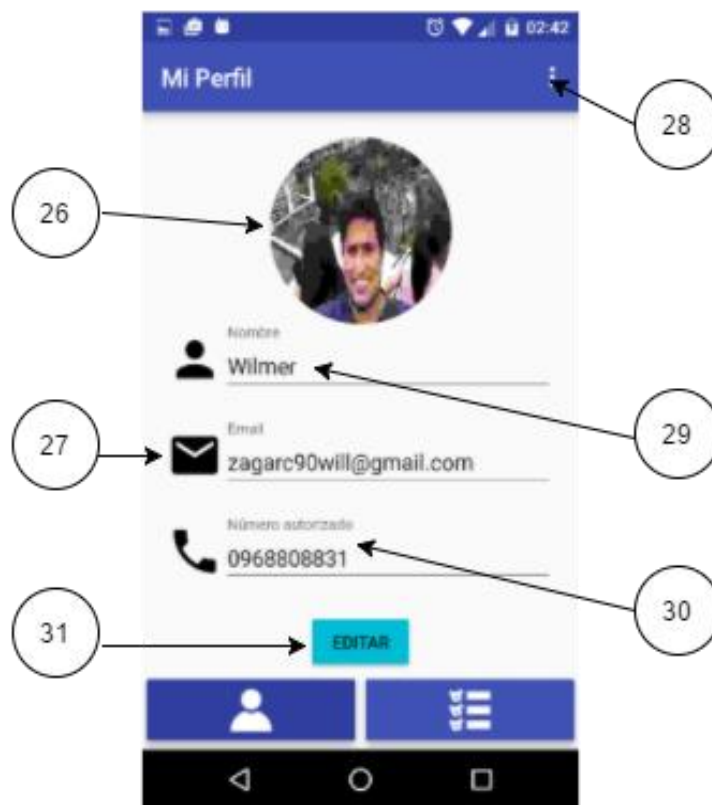
27) Correo del usuario.

28) Botón de menú secundario, aquí se despliega las opciones de “Sobre nosotros” y “Cerrar sesión”.

29) Campo “Nombre de Usuario” que puede ser editado.

30) Campo “Número Telefónico” a cual está atado el dispositivo GPS de la mascota.

31) Botón “EDITAR” el cual ejecuta la acción de editar campos.



32) Componente “ProgressDialog” mientras “busca la mascota (obtiene coordenadas)”



33) Componente de “ProgressDialog” de validación de datos, este componente aparece mientras está validando los datos del usuario.



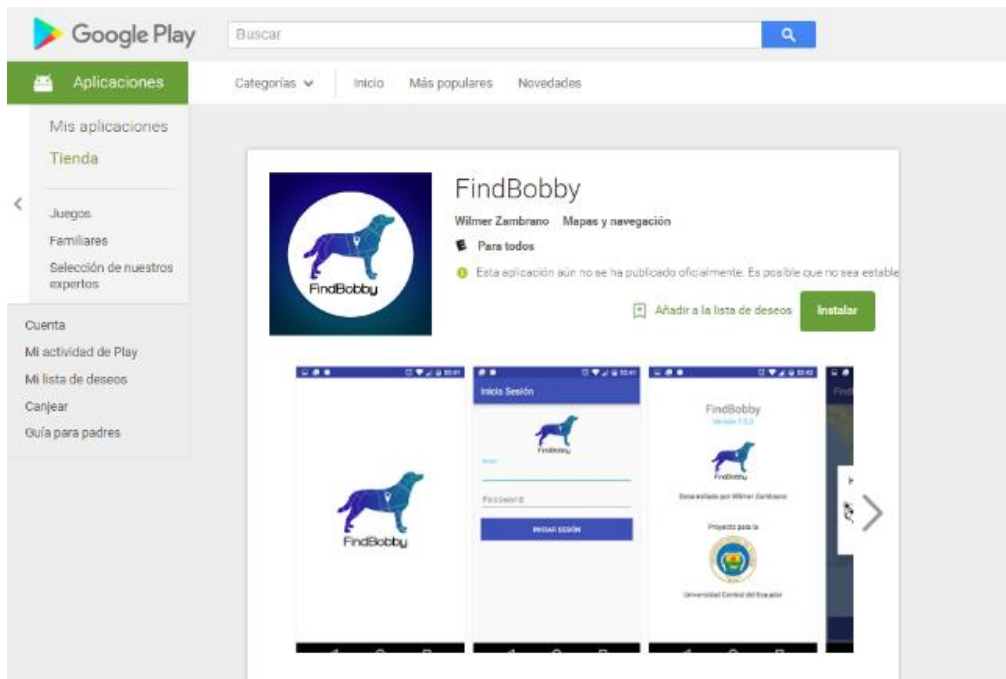
ANEXO E

LIBERACION DE API RESTful y APLICATIVO MOVIL

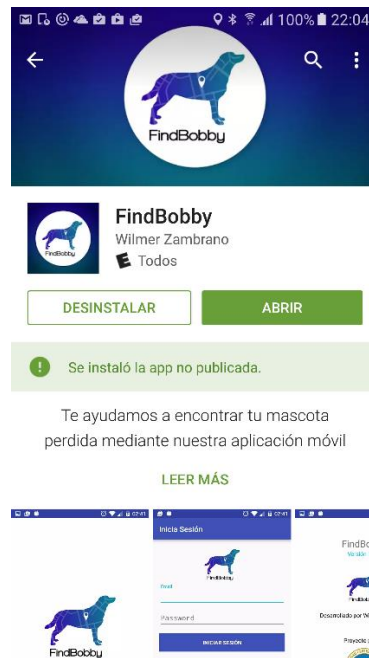
- a) Aplicación móvil desplegada en Google Play la puede encontrar en el siguiente enlace.

<https://play.google.com/store/apps/details?id=com.findbobby.first.firtapp&hl=es>

- 1) Vista de la aplicación desde la PC



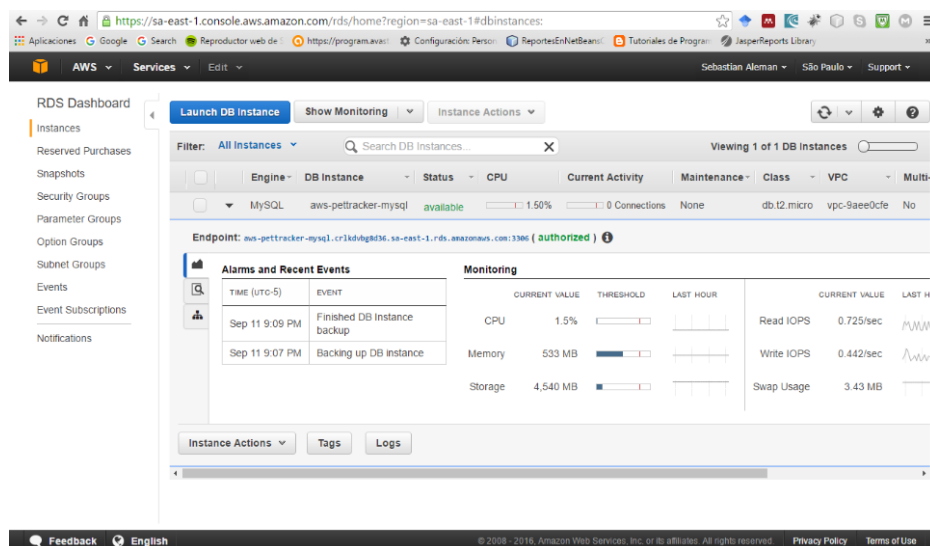
2) Vista de la aplicación desde un dispositivo móvil.



3) Consola de RDS de Amazon:

Ruta de Acceso de contenedor de la base de datos:

aws-pettracker-mysql.crlkdvg8d36.sa-east-1.rds.amazonaws.com:3306



4) Consola de EC2 de Amazon:

Rutas de acceso de contenedor del Servicio Web:

IP Pública: 52.67.34.88

DNS Público: ec2-52-67-34-88.sa-east-1.compute.amazonaws.com

The screenshot displays the AWS Management Console interface for an EC2 instance. The top navigation bar shows the AWS logo, 'Services', and 'Edit'. The left sidebar contains a menu with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area shows a table with one instance, 'pettraacker dev', with ID 'i-089cf5bce54389723', type 't2.micro', and state 'running'. Below the table, the 'Description' tab is selected, showing various instance attributes in a key-value format.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
pettraacker dev	i-089cf5bce54389723	t2.micro	sa-east-1a	running	2/2 checks...	None	ec2-52-67-34-88.sa-east-1.compute.amazonaws.com

Instance: i-089cf5bce54389723 (pettraacker.dev) Public DNS: ec2-52-67-34-88.sa-east-1.compute.amazonaws.com

Description	
Instance ID	i-089cf5bce54389723
Instance state	running
Instance type	t2.micro
Private DNS	ip-172-31-2-149.sa-east-1.compute.internal
Private IPs	172.31.2.149
Secondary private IPs	
VPC ID	vpc-9aee0cfe
Subnet ID	subnet-39a2255d
Network Interfaces	eth0
Source/dest. check	True
Public DNS	ec2-52-67-34-88.sa-east-1.compute.amazonaws.com
Public IP	52.67.34.88
Elastic IPs	
Availability zone	sa-east-1a
Security groups	rds-launch-wizard, mysql-rds, default, launch-wizard-1, view rules
Scheduled events	No scheduled events
AMI ID	amzn-ami-hvm-2016.03.2.x86_64-gp2 (ami-106ee57c)
Platform	-
IAM role	-
Key pair name	pettrack

ANEXO F

Herramientas utilizadas para la generación del prototipo de la aplicación móvil.

- **Material Design:** Framework de diseño para el Front de aplicaciones móviles creada por Google, el cual maneja conceptos de diseño basado en realidad tangible con componentes de diseño interactivos (Acción – Reacción y materialidad de los componentes)
Página oficial: <https://material.google.com/>
- **PSR:** PHP Standards Recommendations el cual es el marco de interoperabilidad publicado por el grupo PHP que sirva para estandarizar los conceptos de programación en PHP.
Página oficial: <http://www.php-fig.org/psr/>
- **Java Standards Android:** Dicta las convenciones y reglas para el estilo de codificación de aplicaciones Android.
Página oficial: <https://source.android.com/source/code-style.html>
- **API RESTful:** Es la implementación de servicio web usando la arquitectura REST (Representational State Transfer) el cual es una arquitectura de desarrollo web basado en los protocolos de comunicación HTTP que no maneja estados, que se apoya en la utilización URIs (Uniform Resource Identifier) bien definidos orientados a manejo de recursos.
- **Android Studio IDE:** Es un entorno de desarrollo integrado para la plataforma Android.
Página oficial: <https://developer.android.com/studio/index.html?hl=es-419>
- **Android SDK Manager:** por sus siglas en inglés: Software Development Kit) es el conjunto de herramientas y librerías desarrolladas por Google para desarrollar, compilar y depurar aplicaciones para el sistema operativo Android

Página oficial: <https://developer.android.com/studio/intro/update.html?hl=es-419>

- **Android Virtual Device AVD:** “Un AVD o Dispositivo Virtual Android es una máquina virtual que ejecuta la plataforma Android y la cual es un perfil de hardware de un dispositivo móvil, y sirve para comprobar la funcionalidad de la aplicación desarrollada” (Revelo, 2014).

Página oficial: <http://www.hermosaprogramacion.com/2014/08/android-programar-app-como/>

- **Vagrant:** una herramienta que nos permite crear cualquier entorno de desarrollo basado en máquinas virtuales. Ofrece una interfaz fácil de usar para crear servidores perfectamente configurados e independientes del sistema operativo del desarrollador.

Página oficial: <https://www.vagrantup.com/>

- **Homestead:** Es el paquete oficial de Vagrant "box" que proporciona un entorno de desarrollo para aplicaciones basado en el Framework Laravel

Página oficial: <https://laravel.com/docs/5.3/homestead>

- **Laravel:** Es un Framework par PHP de código abierto para el desarrollo de aplicaciones web basado en la arquitectura MVC (Cliente – Vista - Controlador)

Página oficial: <https://laravel.com/>

- **Amazon Web Service:** Es una colección de servicios de computación en la nube (también llamados servicios web) que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon.com

Página oficial: <https://aws.amazon.com/es/>

- **Google Play:** Es una plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android, así como una tienda en línea desarrollada y operada por Google

Página oficial: <https://play.google.com/store?hl=es>

- **Trello:** Es un gestor de tareas que permite el trabajo de forma colaborativa mediante tableros (board) compuestos de columnas (llamadas listas) que representan distintos estados.
Página oficial: <https://trello.com/>
- **Git:** Es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente
Página oficial: <https://git-scm.com/>
- **SourceTree:** Es un potente GUI (Graphical User Interface – Interfaz Gráfica de Usuario) para gestionar todos tus repositorios ya sean Git o Mercurial, disponible para winwdos y MAC
Página oficial: <https://es.atlassian.com/software/sourcetree/overview/>
- **Bitbucket:** Es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de revisiones Mercurial y Git.
Página oficial: <https://bitbucket.org/>
- **Json Web Token (JWT):** Método para representar las reclamaciones de forma segura entre dos partes.
Página oficial: <https://jwt.io/>
- **Historia de usuario:** Es una representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario

ANEXO G

Muestra HTML del archivo SPECS

Dirección web del archive specs:

<https://ec2-52-67-34-88.sa-east-1.compute.amazonaws.com/specs.html/>

Session

Create Session Token

Session Error

Session Error

Error User Inactive

Error Token Invalid

Error Token Expired

Error Token something is wr...

User

User Register

User Information

Pets

Pet Register

Pet Information

Pet Update

FindBobby API Specs

Session

CREATE SESSION TOKEN

POST

/session

Create Session Token

Returns user logged.

Example URI

POST /session

Request [Show](#)

Response 200 [Show](#)

Session Error

SESSION ERROR

- Response 401
 - Body

Pets

PET REGISTER

POST

/pets

Pet Register

Example URI

POST /pets

Request

[Hide](#)

Headers

```
Content-Type: "application/json"
```

Body

```
{
  "name": "Dino",
  "age": 1,
  "breed": "Beagle",
  "number": "0981266818",
  "image": "/9j/4AAQSkZJRgABAQEAAQABAA.."
}
```

Response 200

[Hide](#)

Body

```
{
  "message": "Pet created successfully."
}
```

PET INFORMATION

GET

/pets/

Pet Information

Returns user's data

- Parameters:
 - id = `` (int, 4) ... The `id` of the pet.

Example URI

GET /pets/

Request

[Show](#)

Response 200

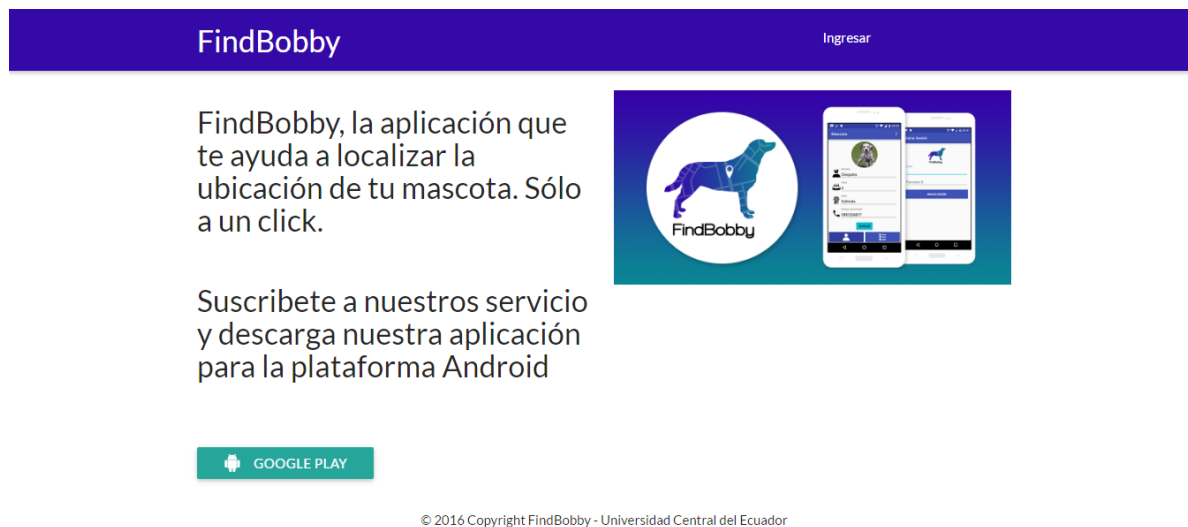
[Show](#)

ANEXO H

Sistema Web

Dirección del sistema web: <http://ec2-52-67-34-88.sa-east-1.compute.amazonaws.com/>


Pantalla principal del sistema con enlaces a sitio de descarga de la aplicación y de inicio de sesión.







Pantalla de inicio de sesión, las credenciales para iniciar sesión son las que se utiliza en la aplicación móvil.



Pantalla de inicio, muestra un listado de las mascotas las cuales pueden ser editadas o eliminadas.


 FindBobby Salir

Mis Mascotas

Nombre	Edad	Raza	Número Móvil Asociado	Acciones
Pluton	5	Dalmata	0997040280	 
LunaL	2	Beagle	0999999999	 

© 2016 Copyright FindBobby - Universidad Central del Ecuador

Pantalla de edición de mi perfil.

 FindBobby Salir

Editar Perfil

Nombre


Wilmer Zambrano.

Número Móvil

0968808831

Email

zagarc9Ovill@gmail.com




IMAGEN

GUARDAR >

© 2016 Copyright FindBobby - Universidad Central del Ecuador

Pantalla de edición de mascotas.



FindBobby

Salir

Editar Mascota

Nombre Pluton	Número Móvil 0997040280
Raza Dalmata	Edad




IMAGEN dog1.jpg

Completa este campo

GUARDAR

© 2016 Copyright FindBobby - Universidad Central del Ecuador