



**UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES**

**CENTRO DE CIENCIAS BÁSICAS**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**TESIS**

**MODELO PARA EL DESARROLLO DE APLICACIONES NATIVAS EN ANDROID  
BASADO EN MEJORES PRÁCTICAS, METODOLOGÍAS ÁGILES Y ELEMENTOS  
DEL ÁREA INTERACCIÓN HUMANO – COMPUTADORA**

**PRESENTA**

**Carlos Alberto Veloz Vidal**

**PARA OBTENER EL GRADO DE MAESTRO EN CIENCIAS**

**TUTOR**

**Dr. Francisco Javier Álvarez Rodríguez (UAA)**

**COMITÉ TUTORAL**

**Dr. Miguel Vargas Martín (UOIT)**

**Dr. Ricardo Mendoza Rodríguez (ITA)**

**Aguascalientes, Ags, 18 de Febrero del 2016**

CARLOS ALBERTO VELOZ VIDAL  
MAESTRIA EN CIENCIAS CON OPCION  
A LA COMPUTACION Y MATEMATICAS APLICADAS  
P R E S E N T E.

Estimado alumno:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los revisores de su trabajo de tesis y/o caso práctico titulado: **"Modelo para el desarrollo de aplicaciones nativas en Android basado en mejores prácticas, metodologías ágiles y elementos del área interacción humano-computadora"**, hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

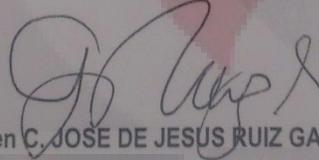
Sin otro particular me permito saludarle muy afectuosamente.

**ATENTAMENTE**

Aguascalientes, Ags., a 18 de febrero de 2016

*"Se lumen proferre"*

**EL DECANO**



M. en C. JOSE DE JESUS RUIZ GALLEGOS



UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES  
FORMATO DE CARTA DE VOTO APROBATORIO

**M. en C. José de Jesús Ruíz Gallegos**  
DECANO DEL CENTRO DE CIENCIAS BÁSICAS  
P R E S E N T E

Por medio del presente como Tutor designado del estudiante **CARLOS ALBERTO VELOZ VIDAL** con ID **110359** quien realizó la tesis titulada: **MODELO PARA EL DE DESARROLLO DE APLICACIONES NATIVAS EN ANDROID BASADO EN MEJORES PRÁCTICAS, METODOLOGÍAS ÁGILES Y ELEMENTOS DEL ÁREA INTERACCIÓN HUMANO – COMPUTADORA**, y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla, y así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE  
"Se Lumen Proferre"  
Aguascalientes, Ags., a 20 de Enero de 2016.

A handwritten signature in black ink, appearing to read 'Francisco Alvarez Rodriguez', written over a faint circular stamp.

Dr. Francisco Javier Alvarez Rodriguez.  
Tutor de tesis

c.c.p.- Interesado  
c.c.p.- Secretaria de Investigación y Posgrado  
c.c.p.- Jefatura del Depto. de Ciencias de la Computación.  
c.c.p.- Consejero Académico  
c.c.p.- Minuta Secretario Técnico



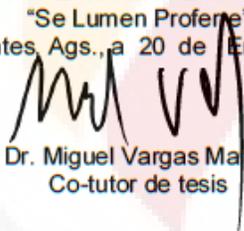
UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES  
FORMATO DE CARTA DE VOTO APROBATORIO

**M. en C. José de Jesús Ruíz Gallegos**  
DECANO DEL CENTRO DE CIENCIAS BÁSICAS  
P R E S E N T E

Por medio del presente como Tutor designado del estudiante **CARLOS ALBERTO VELOZ VIDAL** con ID 110359 quien realizó la tesis titulada: **MODELO PARA EL DE DESARROLLO DE APLICACIONES NATIVAS EN ANDROID BASADO EN MEJORES PRÁCTICAS, METODOLOGÍAS ÁGILES Y ELEMENTOS DEL ÁREA INTERACCIÓN HUMANO – COMPUTADORA**, y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla, y así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

**A T E N T A M E N T E**  
"Se Lumen Profere"  
Aguascalientes, Ags., a 20 de Enero de 2016.



Dr. Miguel Vargas Martín  
Co-tutor de tesis

c.c.p.- Interesado  
c.c.p.- Secretaría de Investigación y Posgrado  
c.c.p.- Jefatura del Depto. de Ciencias de la Computación  
c.c.p.- Consejero Académico  
c.c.p.- Minuta Secretario Técnico



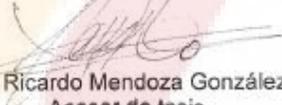
UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES  
FORMATO DE CARTA DE VOTO APROBATORIO

**M. en C. José de Jesús Ruiz Gallegos**  
DECANO DEL CENTRO DE CIENCIAS BÁSICAS  
P R E S E N T E

Por medio del presente como Tutor designado del estudiante **CARLOS ALBERTO VELOZ VIDAL** con ID **110359** quien realizó la tesis titulada: **MODELO PARA EL DE DESARROLLO DE APLICACIONES NATIVAS EN ANDROID BASADO EN MEJORES PRÁCTICAS, METODOLOGÍAS ÁGILES Y ELEMENTOS DEL ÁREA INTERACCIÓN HUMANO – COMPUTADORA**, y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla, y así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

**ATE NTAMENTE**  
"Se Lumen Proferre"  
Aguascalientes, Ags., a 20 de Enero de 2016.

  
**Dr. Ricardo Mendoza González**  
Asesor de tesis

c.c.p.- Interesado  
c.c.p.- Secretaría de Investigación y Posgrado  
c.c.p.- Jefatura del Depto. de Ciencias de la Computación.  
c.c.p.- Consejero Académico  
c.c.p.- Mnueta Secretario Técnico



## **Agradecimientos**

A la Universidad Autónoma de Aguascalientes por haberme formado como profesional, por contribuir en mi desarrollo académico y profesional.

A la University of Ontario Institute of Technologic por las facilidades brindadas durante mi estancia de investigación

Al Dr. Miguel Vargas Martin y al Dr. Ricardo Mendoza Rodríguez por ser parte de mi comité tutorial, su apoyo y acompañamiento a lo largo de este proyecto.

Al Dr. Francisco Javier Álvarez Rodríguez por asesorarme durante la elaboración de este proyecto.



## Dedicatorias

A mis padres Tomas Veloz Santos y Antonia Vidal Ramos por apoyarme en cada etapa de mi vida, por su paciencia y educación.

A mis hermanos Cesar Veloz Vidal y Nayeli Veloz Vidal por su apoyo y comprensión.

A mis compañeros y amigos por su apoyo y solidaridad.





**Contenido**

Agradecimientos..... 7

Dedicatorias ..... 8

Índice de Tablas..... 8

Índice de Gráficas..... 9

Resumen ..... 11

Abstract ..... 12

1. Introducción..... 13

    1.1 Pregunta de investigación..... 13

    1.2 Hipótesis..... 14

    1.3 Objetivo general..... 14

    1.4 Objetivos específicos ..... 14

    1.5 Metodología de la investigación..... 15

2. Aplicaciones nativas en Android..... 16

    2.1 Introducción..... 16

    2.2 Definición de dispositivo móvil..... 17

    2.3 Android ..... 17

        2.3.1 Aplicaciones y su clasificación..... 18

    2.4 Aplicaciones nativas en Android para el tratamiento de problemas de salud..... 20

        2.4.1 Características de las aplicaciones de e-health ..... 21

        2.4.2 Elementos de las interfaces de usuario para aplicaciones de salud ..... 23

3. Mejores prácticas para dispositivos móviles..... 29

    3.1 Mejores prácticas ..... 29

    3.2 Web de desarrolladores de Google ..... 30

4. Mejores prácticas primer acercamiento .....	33
4.1 Mejores prácticas identificadas .....	33
4.2 Evaluación de la primera versión de mejores prácticas identificadas .....	35
4.2.1 Objetivos .....	35
4.2.2 Selección de los participantes .....	35
4.2.3 Equipos .....	35
4.2.4 Aplicaciones .....	36
4.2.5 Instrumentos de evaluación .....	37
4.3 Mejores prácticas segunda versión .....	44
4.4 Productos generados .....	49
5. Herramientas para el diseño de aplicaciones nativas .....	54
5.1 Modelos para el desarrollo de software .....	54
5.2 Ambientes de desarrollo .....	56
5.3 Metodologías ágiles .....	58
6. Solución propuesta .....	61
6.1 Modelo de desarrollo de aplicaciones nativas en Android basado en mejores prácticas, metodologías ágiles y elementos del área interacción humano - computadora .....	63
6.2 Requerimientos .....	64
6.2.1 Instalación y configuración del ambiente de desarrollo .....	65
6.2.2 Levantamiento de los requerimientos .....	66
6.2.3 Implementación del modelo vista – controlador .....	66
6.2.4 Selección de las herramientas de gestión del proyecto .....	67
6.3 Diseño .....	68
6.3.1 Estructura general de la aplicación .....	69
6.3.2 Vista por niveles .....	69

6.3.3 Construcción de las interfaces de usuario .....	70
6.3.4 Navegación e interacción de los elemento de la interfaz de usuario.....	70
6.4 Desarrollo .....	71
6.4.1 Codificación por pares.....	71
6.4.2 Codificación de las clases e interfaces.....	72
6.4.3 Implementación de los frameworks.....	72
6.5 Pruebas.....	73
6.5.1 Creación y ejecución de un caso de prueba .....	74
6.5.2 Probar la funcionalidad de la interfaz de usuario .....	74
6.5.3 Configuración de las unidades virtuales de prueba.....	75
6.5.4 Ejecución en ambientes virtuales.....	75
6.5.5 Configuración de los dispositivos físicos de prueba.....	76
6.5.6 Ejecución en dispositivos físicos .....	77
6.6 Liberación .....	77
6.6.1 Documentación previa al lanzamiento.....	77
6.6.2 Generación del archivo apk.....	78
6.6.3 Publicación .....	79
6.6.4 Retroalimentación de los usuarios .....	79
6.6.5 Manejo de versiones.....	80
7. Caso de estudio .....	81
8. Resultados .....	85
9. Discusión de resultados.....	92
10. Conclusión .....	95
Glosario.....	97
Bibliografía .....	99

Anexos..... 103

Mejores prácticas para el desarrollo de aplicaciones nativas en Android de e-health y su publicación en la plataforma de Google Play..... 102

Desarrollo de aplicaciones de e-health en Android para dispositivos móviles, diagnóstico para Diabetes Mellitus tipo II ..... 110

Desarrollo de Aplicaciones Móviles para e-health: Aplicación Test de Obesidad..... 119

Aplicación Android con fines de salud, bajo el caso de estudio del monitoreo de obesidad ..... 126

Cuestionario utilizado en el caso de estudio..... 134



## Índice de Tablas

<b>Tabla 1.</b> Características y propiedades de aplicaciones de e-health.....	22
<b>Tabla 2.</b> Primera versión de mejores prácticas.....	34
<b>Tabla 3.</b> Resultados del segundo instrumento de evaluación.....	43
<b>Tabla 4.</b> Segunda versión de mejores prácticas. ....	44
<b>Tabla 5.</b> Cuestionario para evaluación del proceso.....	82
<b>Tabla 6.</b> Métricas de medición.....	83



## Índice de Gráficas

<b>Gráfica 1.</b> Pantalla de bienvenida y menú principal.....	25
<b>Gráfica 2.</b> Pantallas para el instrumento de evaluación .....	25
<b>Gráfica 3.</b> Plantillas de las pantallas de resultados, información complementaria y créditos.....	26
<b>Gráfica 4.</b> Instrumento de evaluación.....	38
<b>Gráfica 5.</b> Resultados del instrumento de evaluación .....	38
<b>Gráfica 6.</b> Pregunta uno, primera versión. ....	39
<b>Gráfica 7.</b> Pregunta dos, primera versión.....	40
<b>Gráfica 8.</b> Pregunta tres, primera versión. ....	40
<b>Gráfica 9.</b> Pregunta cuatro, primera versión.....	41
<b>Gráfica 10.</b> Pregunta cinco, primera versión. ....	41
<b>Gráfica 11.</b> Pregunta seis, primera versión. ....	42
<b>Gráfica 12.</b> Pregunta uno, segunda versión. ....	46
<b>Gráfica 13.</b> Pregunta dos, segunda versión. ....	46
<b>Gráfica 14.</b> Pregunta tres, segunda versión. ....	47
<b>Gráfica 15.</b> Pregunta cuatro, segunda versión. ....	47
<b>Gráfica 16.</b> Pregunta cinco, segunda versión. ....	48
<b>Gráfica 17.</b> Pregunta seis, segunda versión. ....	48
<b>Gráfica 18.</b> Pregunta siete, segunda versión.....	49
<b>Gráfica 19.</b> Pantallas de la aplicación Cuestionario sobre depresión .....	50
<b>Gráfica 20.</b> Pantallas de la aplicación Diabetes II .....	51
<b>Gráfica 21.</b> Pantallas de la aplicación Test alcoholismo .....	51
<b>Gráfica 22.</b> Resultados de las aplicaciones publicadas .....	52
<b>Gráfica 23.</b> Modelo propuesto. ....	64
<b>Gráfica 24.</b> Primera pregunta del instrumento de evaluación.....	85
<b>Gráfica 25.</b> Segunda pregunta del instrumento de evaluación. ....	86
<b>Gráfica 26.</b> Tercera pregunta del instrumento de evaluación.....	86
<b>Gráfica 27.</b> Cuarta pregunta del instrumento de evaluación. ....	87
<b>Gráfica 28.</b> Quinta pregunta del instrumento de evaluación. ....	87
<b>Gráfica 29.</b> Sexta pregunta del instrumento de evaluación.....	88

**Gráfica 30.** Séptima pregunta del instrumento de evaluación. ....88  
**Gráfica 31.** Octava pregunta del instrumento de evaluación. ....89  
**Gráfica 32.** Novena pregunta del instrumento de evaluación. ....89  
**Gráfica 33.** Décima pregunta del instrumento de evaluación. ....90  
**Gráfica 34.** Décimo primer pregunta del instrumento de evaluación. ....90  
**Gráfica 35.** Décimo segunda pregunta del instrumento de evaluación. ....91



## Resumen

Hoy en día Android es uno de los sistemas operativos más populares para dispositivos móviles, por los recursos y facilidades para la distribución de aplicaciones. Si bien, existen diferentes estrategias para el desarrollo de aplicaciones para esta plataforma, como el uso de mejores prácticas, disponibilidad de ambientes de desarrollo, metodologías ágiles, etc. Por separado, no constituyen una solución integral para orientar a los desarrolladores a través de las distintas etapas del proceso del software.

A medida que la finalidad de dichas aplicaciones se vuelve más crítica, es necesario considerar diferentes aristas para crear un producto robusto. Por lo tanto, crear un proceso para el desarrollo de aplicaciones nativas en Android, que considere aspectos clave de las áreas interacción humano – computadora, para el diseño de interfaces; metodologías ágiles, para la administración adecuada del proyecto; considere los distintos ambientes de desarrollo, de acuerdo al tipo de producto a generar; y el uso de mejores prácticas, para aumentar la eficacia, calidad, y escalabilidad del software; puede constituir una solución integral al problema de desarrollar aplicaciones para esta plataforma.

Actualmente existen diferentes recursos disponibles en la red para guiar a los desarrolladores, sin embargo, aún no se ha publicado un proceso de ingeniería de software que considere las fases del ciclo de vida del software. Por lo cual, el presente proyecto pretende contribuir a reducir esta brecha.

## Abstract

Nowadays Android is one of the most popular operative systems for mobile devices, its success is due to all the resources available and ease for apps distribution. Although there are different strategies for app development on this platform (e.g. best practices for coding, several Integrated Development Environments (IDE) on web, agile methodologies, etc.) separately, they do not constitute an integral solution for guiding developer through the different stages of software life cycle.

As Android apps are becoming more critical to user is needed to consider another edges in order to develop more comprehensive products. Therefore, designing a process for native Android app development which considers key areas of human – computer interaction, for user interface design; agile methodologies for an appropriated project management; considerate different IDEs according the aim of the app; and using best practices for improving efficiency, quality and scalability of code, all of this can be the basis for designing a solution aimed to breach the gap of developing native Android apps.

At the moment of writing this document, has not been published another model or formerly a software engineering process for guiding the developers through the different stages of software life cycle.

## 1. Introducción

Existen varias alternativas reportadas en la literatura para el desarrollo de aplicaciones móviles. Algunos trabajos como los presentados por Tonini et al. [1], y Oh et al. [2], sugieren el uso de mejores prácticas para el desarrollo de aplicaciones nativas. Otros autores como Thung et al. [3] sugieren el uso de guías y patrones de diseño buscando mejorar la usabilidad, y reusabilidad de las interfaces de usuario en dispositivos móviles. Finalmente, otras estrategias se han reportado en la literatura académica, Corral et al. [4] presenta una extensa revisión de distintas metodologías ágiles como Mobile-D, MASAM, Scrum, metodologías híbridas para el diseño de procesos, entre otras, las cuales dan una mayor prioridad al producto final en lugar de seguir todos los pasos de un proceso tradicional de ingeniería de software.

Por separado cada una de estas estrategias implementa su propia metodología, sus directrices y herramientas, lo cual ha mostrado ventajas y desventajas en su implementación para las distintas plataformas móviles disponibles. Sin embargo, ninguna de estas opciones por si sola constituye una solución integral para el diseño de aplicaciones móviles para una plataforma específica. Como menciona Wasserman [5] a medida que las aplicaciones móviles se vuelven más complejas y de propósitos cada vez más críticos, las organizaciones deben presentar procesos que aborden más aspectos del proceso de desarrollo que son cubiertos hoy en día por los procesos ágiles y en los ambientes de desarrollo.

Por lo tanto, al incorporar las lecciones aprendidas de cada una de estas técnicas dentro de un proceso de desarrollo de aplicaciones puede constituir una solución integral a este problema. Ya que se considerarían elementos dentro del proceso de desarrollo que de manera independiente no son tomados en cuenta, enfocando los esfuerzos en generar un producto robusto centrado en el usuario.

### 1.1 Pregunta de investigación

¿Cuáles serían los elementos a considerar de las metodologías ágiles, mejores prácticas, interacción humano – computadora para conformar un proceso para el desarrollo de aplicaciones nativas en Android?

## 1.2 Hipótesis

H0. No hay un conjunto de mejores prácticas, principios de metodologías ágiles, y elementos del área de interacción humano-computadora que constituyan una base para la construcción de un proceso de desarrollo de aplicaciones nativas para la plataforma Android.

H1. Existe un conjunto no exhaustivo de mejores prácticas, principios de metodologías ágiles, y elementos del área de interacción humano-computadora que constituyen una base para la construcción de un proceso de desarrollo de aplicaciones nativas para la plataforma Android.

## 1.3 Objetivo general

Definir un proceso de desarrollo para la construcción de aplicaciones nativas en Android basado en elementos de mejores prácticas, metodologías ágiles, e interacción humano-computadora para la plataforma Android.

## 1.4 Objetivos específicos

- Revisión de la literatura sobre los tipos de aplicaciones, y las mejores prácticas dentro de la plataforma Android
- Construcción de una aplicación en Android bajo un enfoque de conocimiento empírico sobre esta plataforma, e identificación de las mejores prácticas para el desarrollo de aplicaciones nativas
- Realizar un caso de estudio sobre las mejores prácticas identificadas, analizar los resultados obtenidos y las áreas de oportunidad
- Revisión de la literatura sobre las estrategias y herramientas para la construcción de aplicaciones nativas en dispositivos móviles
- Identificación de los beneficios para diseñadores, desarrolladores y usuario final
- Diseño del proceso para el desarrollo de aplicaciones nativas para la plataforma Android

- Medir la percepción de los estudiantes sobre el proceso diseñado en cuanto a usabilidad, facilidad de uso, y aceptación
- Aceptación o rechazo de la hipótesis

### **1.5 Metodología de la investigación**

La primera parte de la investigación es un análisis y revisión de la literatura sobre los tipos de aplicaciones que existen y las mejores prácticas para su construcción. Bajo un enfoque empírico, se desarrollara una aplicación, lo cual permitirá identificar un conjunto no exhaustivo de mejores prácticas que no han sido reportadas en la literatura. Posteriormente, se diseñará un primer caso de estudio para evaluar las prácticas identificadas, y sus áreas de oportunidad como una solución a la problemática identificada.

De nueva cuenta se hará una revisión de la literatura sobre las herramientas y estrategias para el desarrollo de aplicaciones nativas en dispositivos móviles, con el objetivo de identificar los elementos esenciales de las herramientas para la construcción de aplicaciones, los fundamentos de las metodologías ágiles, y del área interacción humano-computadora. De manera que sea posible integrar estos elementos junto con las mejores prácticas identificadas previamente, para diseñar un proceso para el desarrollo de aplicaciones nativas en Android. El cual será evaluado por estudiantes con conocimientos en: programación orientada a objetos, diseño de sitios web adaptados para dispositivos móviles, Android, y procesos de ingeniería de software, para medir la usabilidad, facilidad de uso, y aceptación de la propuesta a diseñar.

## 2. Aplicaciones nativas en Android

Existen diferentes tipos de aplicaciones que se pueden desarrollar en Android. Algunas de ellas en función del dispositivo móvil donde se ejecutaran como Smart TVs, smartphones, tablets, Smart-watch, etc. Y a su vez cada una de ellas tiene sus propiedades y comportamientos definidos. Esto es posible gracias a la aceptación del sistema operativo Android y la amplia gama de dispositivos en los que se puede instalar.

### 2.1 Introducción

Los dispositivos móviles se han convertido en una parte esencial en la vida de las personas. Han pasado de ser simples herramientas que nos permitían realizar llamadas o enviar mensajes de texto, de ser dispositivos monótonos y de poca personalización a ser ahora dispositivos que nos permiten acceder a la red, descargar todo tipo de contenido digital como libros, música, herramientas para el trabajo, la escuela, el hogar, dispositivos que son totalmente personalizables al gusto de cada usuario a tal grado que son dispositivos personales, ya que son una extensión del usuario, definen su identidad y son portadores de su información personal.

Estos objetos tienen características únicas que los distinguen de cualquier otro artículo electrónico que pudiera considerarse dispositivo móvil, ya que son portables, pues es fácil que el usuario los lleve consigo a donde quiera que vaya, no ocupan mucho espacio, además, cuando se termina la batería de estos no es necesaria reemplazarse como anteriormente se hacía con pilas alcalinas, sino que puede conectar el dispositivo a una computadora o por medio de un adaptador a una toma de energía eléctrica para cargarlo y continuar usándolo; como ya se mencionó son dispositivos personales, el usuario confía en ellos información que no comparte con otras personas desde contraseñas, acceso a sus cuentas bancarias, redes sociales, fotografías, videos, etc.; el usuario confía demasiada información en estos dispositivos ya que son de fácil manejo, no son complicados de usar ni se requiere de un conocimiento exhaustivo para su utilización; esta última característica es la que permite que puedan acceder a cualquier conexión a internet disponible y seguir "conectado" con el mundo exterior.

La definición de dispositivo móvil ha cambiado con el paso del tiempo al mismo ritmo que los avances tecnológicos, es por ello que varios dispositivos cabrían dentro de la categoría de dispositivos móviles como consolas de videojuegos portables, reproductores de música, dispositivos para leer libros electrónicos, por mencionar algunos ejemplos. Entonces es recomendable definir que se considera dispositivo móvil para los fines prácticos del presente trabajo.

## **2.2 Definición de dispositivo móvil**

Dejando de lado al fabricante o la marca, ya que existen diversos fabricantes que producen una variedad considerable de dispositivos en distintas presentaciones y tamaños, al igual que el sistema operativo que utilizan, algunos de ellos cuentan con registros y derechos de autor por lo que no se tiene acceso al código fuente y por lo tanto no pueden ser modificados, de igual manera existen versiones libres de sistemas operativos disponibles para ser descargados y editados lo que genera una amplia gama de sistemas.

Un dispositivo móvil puede ser un smartphone (teléfono inteligente) o una tablet, el cual indistintamente de su tamaño debe contar con características mencionadas anteriormente que son portabilidad, conectividad (Wireless LAN (WLAN), 3G/4G), de uso personal (un solo usuario), dispositivos capaces de realizar llamadas y recibir mensajes de texto (característica disponible generalmente solo en smartphones), su sistema operativo debe ser multitarea y debe contar con soporte touch (toque), tener sensores básicos como cámara capaz de grabar video, acelerómetro, sensor de proximidad, sistema de posicionamiento global (GPS por sus siglas en inglés), finalmente deben permitir al usuario descargar nuevas aplicaciones o programas a través de un medio de distribución oficial.

Estas características nos permiten discernir aquellos dispositivos que funcionan para una tarea específica para la cual fueron diseñados, de los dispositivos móviles que son herramientas integrales que satisfacen las necesidades de contenido digital y exigencias de conectividad del usuario.

## **2.3 Android**

La popularidad de los dispositivos móviles ha aumentado en los últimos años, a tal grado que la mayoría de la población posee uno de estos y lo lleva siempre consigo. En México y América Latina la mayoría de estos dispositivos operan con el sistema operativo Android, el cual permite descargar aplicaciones gratuitas, libros, videos y demás contenido digital desde su plataforma de distribución oficial Google Play.

De acuerdo con Firtman [6], Android es un sistema operativo de código abierto basado en Linux desarrollado por Open Handset Alliance, un grupo de compañías lideradas por Google. Al ser de código abierto permite que usuarios puedan descargar el código fuente, modificar el kernel y generar versiones más personalizadas de este sistema operativo, lo que propicia que existen diferentes versiones disponibles en la red.

Google ha realizado un esfuerzo considerable por documentar la mayor parte de este sistema, así como en generar un sitio web dedicado a los desarrolladores donde pueden encontrar información relacionada con Android, este tema se abordara con más detalle en el tercer capítulo.

### 2.3.1 Aplicaciones y su clasificación

Las aplicaciones en Android se desarrollan en lenguaje Java, por lo cual este sistema operativo está basado en la Programación Orientada a Objetos (POO). Otra característica importante de las aplicaciones móviles, de acuerdo con Kraemer [7], es que las aplicaciones reaccionan a eventos y entradas realizadas por el usuario a través de las diferentes interfaces del dispositivo móvil, como por ejemplo los sensores o toques en la pantalla táctil.

Existen diferentes tipos de aplicaciones móviles independientemente del dispositivo donde se ejecutaran como pueden ser teléfonos inteligentes, tabletas, relojes inteligentes, televisores inteligentes, etc. De acuerdo con lo presentado por Vique [8], y Angulo [9], no existe una clasificación formal del tipo de aplicaciones móviles, tampoco la hay en la web de desarrolladores de Android. Por lo que, se propone la siguiente clasificación para aplicaciones móviles en Android con base en su funcionalidad.

*Aplicaciones básicas*, tienen un nivel de interacción básico con el usuario, por lo general permiten la gestión del envío y recepción de mensajes de texto (SMS o MMS). Su funcionamiento es sencillo para los usuarios, de bajo tamaño de instalación (la mayoría

viene pre-cargada en el dispositivo), no requieren de formato especial ni acceso a los sensores del dispositivo móvil.

*Aplicaciones Web o Web móviles*, aquellos sitios web que se han adaptado para que su información y contenido pueda ser visualizado en dispositivos de menor resolución de pantalla a diferencia de laptops y computadoras de escritorio tradicionales. Se desarrollan con el uso de tecnologías web como HTML, CSS, Javascript, al igual que un sitio web regular; la diferencia radica en que este tipo de aplicaciones tampoco puede acceder al uso de los sensores ni cualquier otro periférico propio de los dispositivos móviles.

*Aplicaciones nativas*, aplicaciones que se desarrollan para una plataforma en específico, y para ello se utiliza el SDK oficial. Para el caso de Android se hace uso del Android SDK. No existe ningún tipo de estandarización ya que se utiliza por lo general, el estándar de codificación en función del lenguaje que emplea el sistema operativo. Tienen acceso a las capacidades del ambiente de desarrollo, así como a los sensores y periféricos de un dispositivo móvil y en general un mejor desempeño en cuanto a manejo de código. Por lo tanto, el acceso a los periféricos y sensores del dispositivo junto con las herramientas de desarrollo del SDK oficial, permiten generar aplicaciones más completas y con mejoras en las áreas de desempeño, seguridad, consumo de energía, permisos y restricciones, entre otras.

*Aplicaciones híbridas*, también conocidas como web sobre nativas. Son aplicaciones que no necesitan estar instaladas en el dispositivo móvil para ejecutarse. Están basadas en tecnologías web y se ejecutan ya sea a través de un navegador web, o bien de un servicio web. Su arquitectura consiste en instalar un cliente en el dispositivo y con el uso de un servicio web acceder a la información que la aplicación necesita en el servidor, lo cual le permite acceder a algunos sensores como la cámara y otras funcionalidades del dispositivo móvil.

Sin importar de qué tipo de aplicación se trate, existen diferentes ambientes de desarrollo y herramientas que facilitan su construcción, lo cual permite crear aplicaciones tan complejas y elaboradas como se requiera que sean. A diferencia del software desarrollado de manera tradicional, en los dispositivos móviles apenas se requiere de un equipo de trabajo menor. Por lo general están conformados por cinco personas, donde

no se requiere llevar un proceso formal de documentación dentro del ciclo de desarrollo. Otro aspecto a considerar, es el corto tiempo que se tiene desde la concepción de la idea hasta su liberación como un producto terminado en el mercado, apenas un par de meses; esto debido a la gran demanda de este tipo de productos y su componente de innovación que dura apenas poco tiempo entre los usuarios.

#### **2.4 Aplicaciones nativas en Android para el tratamiento de problemas de salud**

De acuerdo a la Organización Mundial de la Salud (OMS), e-health se define como *"El empleo de información y tecnologías de comunicación para un mejor control de la salud. Por ejemplo, para el tratamiento de determinados pacientes, fomentar la investigación, crear herramientas para la educación de estudiantes, hacer detección de diversas enfermedades, y en fin, para la supervisión de la salud pública"*.

El uso de los dispositivos móviles se ha incrementado a tal grado, que se han convertido en una alternativa al cómputo tradicional, haciendo más simple la movilidad, usabilidad y portabilidad de estos dentro de la población en general, lo cual los convierte en una valiosa plataforma para el desarrollo de aplicaciones relacionadas con temas de salud, debido a:

1. La adopción generalizada de dispositivos móviles (teléfonos inteligentes y tabletas) con capacidades técnicas cada vez más potentes, según datos reportados por Pew Internet & American Life Project [10].
2. La tendencia de las personas a llevar estos dispositivos con ellos a todas partes.
3. De acuerdo con Venta et al. [11] el apego de las personas a sus teléfonos inteligentes.
4. Las características de sensibilización de contexto habilitadas a través de los sensores y la información personal vía telefónica.

La dependencia de las personas a llevar siempre consigo un dispositivo móvil, favorece que instituciones de salud, empresas, y desarrolladores independientes construyan aplicaciones que estén enfocadas a atender problemáticas de salud específicas. A este tipo de productos se les conoce como aplicaciones de e-health.

#### 2.4.1 Características de las aplicaciones de e-health

De manera similar al problema de la clasificación de aplicaciones móviles, no se tienen reportadas en la literatura las características básicas de aplicaciones de e-health. Por lo tanto, se realizó un estudio comparativo de distintas aplicaciones relacionadas a tratar la problemática de obesidad. Se seleccionaron las aplicaciones más descargadas por los usuarios, y se analizaron sus características técnicas. Adicionalmente, se propusieron otras características para complementar un conjunto no exhaustivo de características básicas para aplicaciones de e-health, el cual fue reportado en la literatura por Veloz et al. [12].

Dichas características se mencionan a continuación:

1. *Espacio de instalación mínimo*, el archivo con extensión apk (android Application package) a generar, no debe pesar demasiados Mb para que resulte ser un producto atractivo para descargar entre los usuarios. Una aplicación que requiere de un tamaño de instalación "grande" no es recomendable considerando el espacio de almacenamiento limitado con el que cuentan los dispositivos móviles.
2. *Libre de publicidad*, este tipo de aplicaciones no deben contener ningún tipo de publicidad, ni cualquier otro distractor que desvíe la atención del objetivo original para el que fue diseñada la aplicación. El uso de publicidad suele estar asociado con los videojuegos de descarga gratuita, y se utilizan para recaudar fondos para el equipo desarrollador. En el caso de aplicaciones de e-health el objetivo principal es y debe ser, difundir la concientización de determinado padecimiento entre los usuarios.
3. *Contar con respaldo científico*, de acuerdo con Hesse & Shneiderman [13], las aplicaciones de e-health deben integrar las mejores evidencias de las ciencias de la salud y las ciencias del área de ingeniería de software. Por lo tanto, los instrumentos que se utilicen deben estar validados por alguna institución médica reconocida, o bien, deben ser parte de los productos liberados de una investigación en curso de cualquier universidad, siempre y cuando se haya comprobado de manera cualitativa la eficacia de dichos instrumentos por profesionales del área de la salud. Adicional a esto, es importante incluir la leyenda "Los resultados, diagnósticos, e información mostrada en esta aplicación no pretende reemplazar la información de un experto, al contrario, pueden ser de utilidad para complementar los tratamientos tradicionales". Esta información debe

ser mencionada en la ficha de descarga de Google Play, de esta manera los usuarios podrán verificar la validez de la aplicación y que esta no representa la opinión o consejos del equipo de desarrollo.

4. *Gratuita*, a diferencia del punto número dos, existen aplicaciones de e-health que tienen un costo de descarga y son libres de publicidad. El objetivo principal de las aplicaciones de salud es promover la cultura de la concientización entre los usuarios al hacerles llegar información y/o consejos de manera oportuna, que les permitan tener una mejor calidad de vida. Para lograr esto las aplicaciones deben distribuirse de manera gratuita, al contrario de una aplicación que tiene un precio para descargar la finalidad de estas aplicaciones no es cobrar por información que ha sido compartida y publicada en revistas de divulgación, congresos y organizaciones de salud.

5. *Interfaz limpia*, la interfaz de usuario solo debe contener aquellos elementos que le permitan precisamente al usuario concentrarse en la tarea que está realizando es ese momento, como contestar un cuestionario, o ingresando información de carácter personal sobre su estado de salud, cualquier elemento adicional como publicidad, enlaces a sitios de interés o cualquier otro elemento puede distraer al usuario del objetivo principal de esta familia de aplicaciones.

De acuerdo con Norris et al. [14], la naturaleza penetrante de los dispositivos móviles, especialmente los celulares, puede impulsar el bienestar al orientar y reforzar el mensaje de salud de una manera conveniente. Ellos proponen una estrategia, la cual resume los hallazgos encontrados al asociar los factores de éxito con tareas clave, y dividirla en tres etapas: identificar aplicaciones adecuadas, canalizar las actividades de desarrollo, y confirmar las actividades para sustentabilidad.

**Tabla 1.** Características y propiedades de aplicaciones de e-health

<b>Características identificadas</b>	<b>Propiedades</b>
<b>Espacio de instalación mínimo</b>	Hacer eficiente usos de los recursos
<b>Libre de publicidad</b>	Enfatizar el cuidado de la salud y principios socioeconómicos
<b>Contar con respaldo científico</b>	Promover el bienestar y estilo de vida saludable
<b>Gratuita</b>	Son de un costo efectivo para el proveedor y el consumidor
<b>Interfaz limpia</b>	Son fáciles de usar y convenientes, requie-

En la Tabla 1, se comparan las características identificadas con la estrategia desarrollada por Norris et al. [14], lo que permite identificar las propiedades de cada una de ellas. Adicionalmente, las aplicaciones de e-health pueden ser de cualquiera de los 4 tipos mencionados en la sección anterior, la ventaja del uso de aplicaciones nativas en Android radica en el acceso a los sensores y en cómo a través de su implementación se puede recabar información del usuario.

#### 2.4.2 Elementos de las interfaces de usuario para aplicaciones de salud

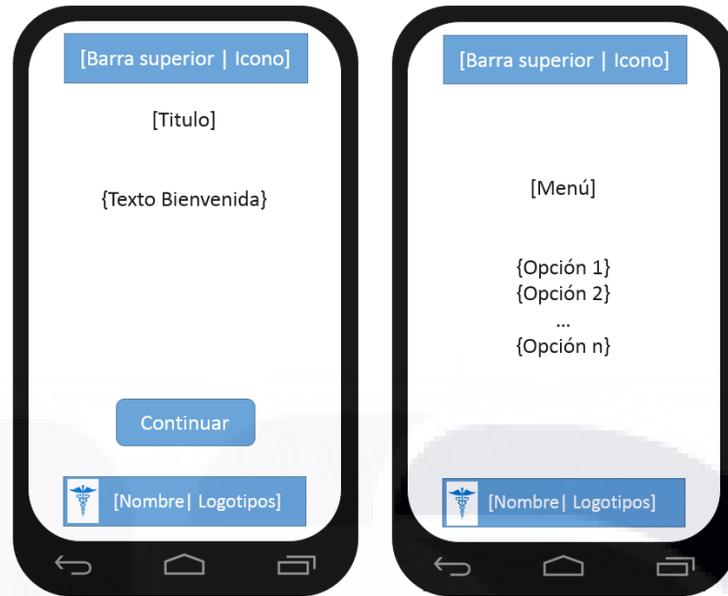
Las interfaces de usuario esta familia de aplicaciones deben ser específicas para los dispositivos donde se van a ejecutar, sin embargo, no se cuentan con guías, patrones de diseño y/o estándares que permitan lograr este objetivo. De manera adicional, las aplicaciones de e-health deben servir para mejorar la calidad de los usuarios que las utilizan.

Existen algunos elementos para el diseño de interfaces en comunidades virtuales de e-health presentados por Mendoza-González et al. [15], que pueden ser adaptados para el desarrollo de aplicaciones en dispositivos móviles como se presenta a continuación.

- Correcta navegación: usar los elementos dentro de la Application Programming Interface (API), junto con los botones físicos del dispositivo para permitir una correcta navegación y transición entre las pantallas de la aplicación. Un ejemplo de esto es el uso de botones que le permitan avanzar hacia la siguiente pantalla o modulo, o bien, el uso del botón de retroceso del dispositivo móvil, que permita regresar a la pantalla anterior cuando esto sea válido.
- Distribución de los elementos, cuando se trabaja con un instrumento de evaluación como un cuestionario con respuestas de opción múltiple es recomendable utilizar radio buttons para indicar cada una de las respuestas disponibles, así como también dejar el suficiente espacio entre cada una de ellas, de esta manera el usuario podrá seleccionar la opción deseada sin que estos elementos se encimen unos con otros por una mala distribución. Se recomienda utilizar el botón de retroceso del dispositivo para regresar a la pregunta anterior en caso de

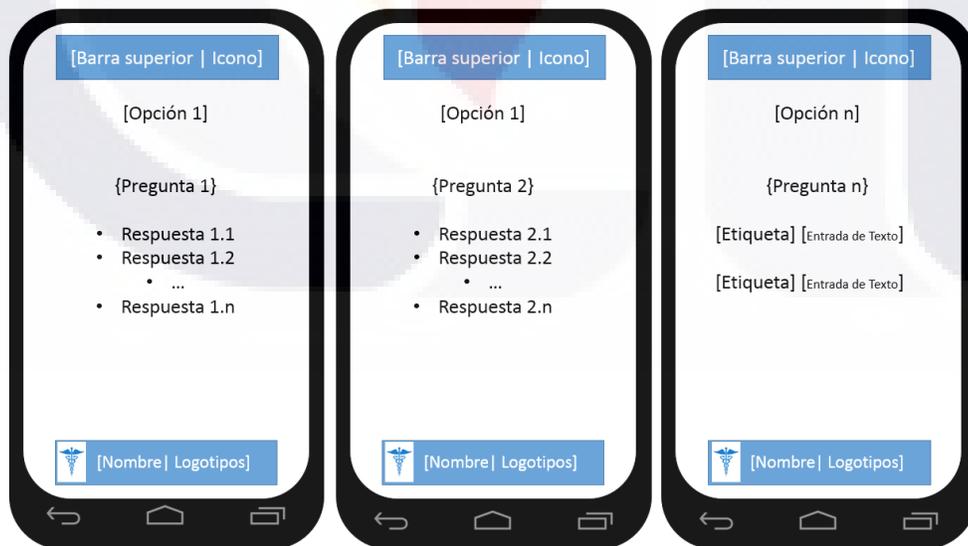
que el usuario accidentalmente haya seleccionado otra opción que no fuera la que él deseaba. Al fijar cada uno de los elementos en una misma posición, brinda el mismo desempeño sin importar la resolución de pantalla o el dispositivo donde este instalada la aplicación.

- Seguridad, las aplicaciones de e-health pueden requerir del acceso a los sensores del dispositivo móvil para recabar información del usuario, así también es posible que los resultados se guarden en archivos de manera local, en una base de datos o bien a través de servicios web. En el escenario más simple no se almacena información alguna del usuario, y si se llega a solicitar información como el peso o estatura es solo para poder conformar el resultado el cual se pierde al momento de cerrar la aplicación. En cualquiera de estos escenarios se debe notificar al usuario a través de un mensaje sobre los permisos que pueda requerir la aplicación para el manejo de información personal.
- Número de pantallas y su distribución: se sugiere que las aplicaciones cuenten con seis pantallas principales, hay ciertos elementos que tienen en común estos son la barra superior, la cual puede contener el icono de la aplicación junto con su nombre, además de la barra inferior, que contiene un icono alusivo a salud, el nombre de la aplicación, y los logotipos de las instituciones que la respaldan. La distribución de las pantallas es la siguiente:
  1. Pantalla de bienvenida, se incluye un título y un mensaje de bienvenida que se utiliza para explicar brevemente el contenido y finalidad de la aplicación que se acaba de descargar así como los instrumentos utilizados y un botón de continuar para avanzar a la siguiente pantalla.
  2. Menú principal, se muestran los diferentes módulos que conforman la aplicación que son el acceso al instrumento de evaluación, un módulo de resultados, otro de consejos y/o información adicional y finalmente los créditos. En la Gráfica 1, se muestra una plantilla para estas dos primeras pantallas.



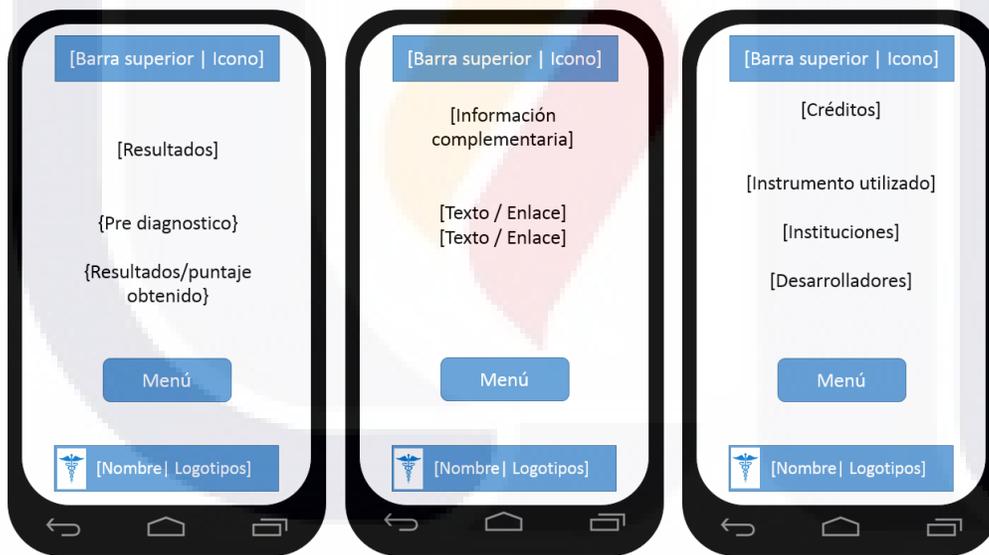
**Gráfica 1.** Pantalla de bienvenida y menú principal

3. Instrumento de evaluación, aquí se muestran los reactivos del instrumento seleccionado, pueden ser preguntas de opción múltiple o bien, campos en los que se pide ingreso de información del usuario como su peso o estatura, por mencionar algunos ejemplos. En la Gráfica 2, se aprecia las plantillas para el instrumento de evaluación.



**Gráfica 2.** Pantallas para el instrumento de evaluación

4. Resultados, aquí se muestran los resultados que obtuvo el usuario después de completar cada uno de los reactivos del instrumento de evaluación seleccionado, se indica el puntaje obtenido o una breve explicación de la situación actual de salud que presenta el usuario, además, se incluye un botón que le permita regresar al menú principal.
5. Información complementaria, aquí se muestra información que pueda resultar de utilidad para el usuario en base a los resultados obtenidos. Pueden ser enlaces, consejos o información que le permitan entender mejor la situación en la que se encuentra y que puede hacer para mejorar su estado de salud.
6. Créditos, información sobre el instrumento utilizado y la institución de donde proviene, se incluye información sobre las instituciones y desarrolladores que crearon la aplicación junto con un botón para volver al menú principal. En la Gráfica 3, se muestran las plantillas correspondientes a las pantallas de resultados, información complementaria y créditos respectivamente.



**Gráfica 3.** Plantillas de las pantallas de resultados, información complementaria y créditos

Como se muestra en Gráfica 1, Gráfica 2, y Gráfica 3 se recomienda trabajar con colores claros (blanco y azul), de manera que transmitan una sensación de calma y permitan al usuario enfocarse en el contenido de la aplicación. Los colores de los textos deben contrastar con los colores utilizados en el fondo y los botones. El uso de estas plantillas puede adaptarse a cualquier contexto y problemática de salud, de manera que al

TESIS TESIS TESIS TESIS TESIS

utilizarlas permiten generar aplicaciones con una misma identidad, con características y propiedades definidas.

Plantillas similares pueden ser encontradas en sitios web de desarrollo para Google, Apple, entre otros. Al integrar diferentes elementos como un diseño minimalista, amigable, consistente, usable no solo para los usuarios también para los desarrolladores, junto con las características y propiedades mencionadas anteriormente, permite que estas plantillas se puedan adaptar a cualquier contexto. El uso de estas plantillas se mostrara más adelante en el cuarto capítulo, las cuales fueron utilizadas para diseñar las interfaces de las aplicaciones que tratan problemáticas sobre depresión, obesidad, alcoholismo, y diabetes.

Finalmente, se debe aclarar hacia qué tipo de usuarios va dirigida la aplicación si son para usuarios con características específicas o usuarios en general; detallar de donde se obtiene el instrumento de evaluación para el padecimiento seleccionado, es decir, si proviene de una investigación en curso o fue publicado por alguna organización de salud; indicar brevemente el contexto social y/o cultural donde se desarrolla la aplicación, al distribuirse en una plataforma global, debe identificarse bajo qué circunstancias la aplicación fue desarrollada, ya que existen varios factores socioeconómicos, demográficos, sociales, políticos, etc. que cambian de país a país; por último, señalar que "Los resultados, diagnósticos, e información mostrada en esta aplicación no pretende reemplazar la información de un experto, al contrario, pueden ser de utilidad para complementar los tratamientos tradicionales".

En resumen, en este capítulo se menciona la situación actual del sistema operativo Android, la problemática existente en cuanto a la categorización de las aplicaciones, y se propone una categorización en base a la funcionalidad de las aplicaciones que además es independiente del ambiente de desarrollo utilizado. Por otra parte, se analiza la situación de las aplicaciones de e-health mencionando sus características principales, y las ventajas que se obtienen al desarrollarse como aplicaciones nativas. De acuerdo con información de un reporte de Pew Internet & American Life Project [10], y Venta et al. [11], se debe aprovechar que los dispositivos móviles cada vez cuentan con mejores características técnicas, además, de que los usuarios tienden a llevar consigo estos dispositivos en todo momento.

Finalmente, con las características presentadas para aplicaciones de e-health, que además fueron reportadas en la literatura por Veloz et al. [12], junto con algunos elementos de comunidades virtuales de e-health reportados por Mendoza-Gonzalez et al. [15] se proponen guías para el diseño de las interfaces de usuario tomando en cuenta elementos de usabilidad como la navegación, distribución, uso de elementos dentro de la interfaz de usuario. Al empatar las características generales de este tipo de aplicaciones con aplicaciones nativas, nos permite considerar a las aplicaciones de e-health como un producto dentro de la categoría de aplicaciones nativas en Android, independiente de las estrategias y herramientas utilizadas para su desarrollo.



### 3. Mejores prácticas para dispositivos móviles

Las mejores prácticas son un conjunto de acciones adoptadas por desarrolladores, ya que su uso se refleja directamente en mejor calidad del producto final, cumplimiento del calendario, uso eficiente de los recursos, entre otros beneficios. A diferencia del cómputo tradicional, los dispositivos móviles tienen mayores limitantes como se ha mencionado. Por lo cual las mejores prácticas para dispositivos móviles también deben adaptarse a las necesidades cambiantes de este ecosistema.

#### 3.1 Mejores prácticas

Las aplicaciones desarrolladas para dispositivos móviles como Android, difieren de los sistemas de software tradicionales por factores como: la importancia que adquieren los requerimientos no funcionales para el correcto desempeño de la aplicación, la funcionalidad del producto final sobre la documentación de cada una de las etapas durante el proceso de desarrollo, el corto tiempo disponible para crear y liberar el producto, las distintas metodologías disponibles tanto para dispositivos móviles llamadas metodologías ágiles como las metodologías convencionales de desarrollo de grandes sistemas de software, por mencionar algunas de las principales diferencias.

Por otra parte, también hay aspectos en común como el uso de mejores prácticas en las distintas fases del ciclo de desarrollo, para asegurar que el producto se entregue en tiempo, que cumpla con los requisitos de funcionabilidad, portabilidad, desempeño entre otros aspectos de calidad.

De acuerdo con Withers [16], una mejor práctica puede ser una práctica técnica o de gestión, que ha demostrado una mejora sobre uno o más aspectos como: productividad, coste, calendario, calidad o satisfacción del usuario.

Un artículo de IBM [17], menciona que la mayoría de los proyectos de software tienden a fracasar debido a que sobrepasan el presupuesto, se demoran, el proyecto aún no está completo, etc. Además, el 30 % de los proyectos de software son pobremente ejecutados por lo que son cancelados antes de ser completados. Esto involucra proyectos que utilizan tecnologías modernas basadas en Java, J2EE, XML, y servicios web,

TESIS TESIS TESIS TESIS TESIS

tomando en cuenta que Android combina el uso de estas tecnologías, no se encuentra exento de esta regla.

Teniendo en cuenta que Android es un lenguaje de programación basado en Java y por lo tanto en programación orientada a objetos, se podría pensar que se pueden tomar las mejores prácticas de este paradigma como se aplican en el software para equipos de cómputo tradicionales y emplearlas para el desarrollo de aplicaciones nativas.

En la práctica se ha demostrado que no es así, debido a que las mejores prácticas utilizadas están dirigidas a proyectos de software grandes, con varias personas trabajando para su desarrollo y mantenimiento, teniendo a su disposición más recursos físicos como la memoria, el espacio de almacenamiento y consumo de energía en comparación con las limitadas capacidades de un dispositivo móvil.

Gran parte del trabajo que se encuentra disponible en la literatura está orientado a resolver problemas de seguridad, integridad y confidencialidad de la información sensible del usuario y/o identificar amenazas o riesgos para el sistema operativo, de acuerdo con lo reportado por Tonini et al. [1], y Oh et al. [2]. Este tipo de trabajos han contribuido a mejorar la seguridad en general de los dispositivos móviles, recordando que el usuario confía en ellos información personal que no desea compartir con alguna otra persona.

De manera adicional un estudio realizado por Infotec [18], en América Latina permitió identificar que existen mejores prácticas que fomentan el desarrollo y consumo de aplicaciones en todos los niveles de la sociedad, sin embargo, dichas estrategias están orientadas hacia el punto de vista comercial.

### **3.2 Web de desarrolladores de Google**

Google ha realizado un esfuerzo considerable al poner al alcance de los desarrolladores una página web dedicada a ello. La web de desarrolladores de Android como se le conoce, cuenta con información sobre el SDK oficial, guías de diseño de una aplicación, cómo construir o integrar módulos y/o funcionalidades a una aplicación, publicar el producto en Google Play, comercializar, herramientas, referencias y demás contenido.

TESIS TESIS TESIS TESIS TESIS

La documentación que se ahí se presenta, describe en detalle cada uno de los elementos disponibles dentro de la API de Android, sus métodos, atributos y propiedades; e incluso se presentan fragmentos de código para mostrar su implementación.

Es posible encontrar información similar en línea, desde una variedad de SDKs, incluso libros para familiarizarse con el lenguaje que no difieren mucho de la información original publicada en la web de desarrolladores de Android.

En general, tanto los libros como la documentación oficial no muestran el camino a seguir para desarrollar una aplicación de manera adecuada, ya que la información disponible se puede asimilar al contenido de una enciclopedia ya que al describir cada uno de los elementos de la API por separado. De acuerdo con Robillard [19] esta estrategia no provee una imagen general de cómo interactúan los diferentes componentes, además los códigos de ejemplo pueden resultar muy sencillos y su nivel de complejidad no se compara con el nivel que se pretende alcanzar en la aplicación a desarrollar. Estos problemas se presentan en desarrolladores con experiencia, y se hacen más evidentes en personas que desean crear una aplicación en Android pero no saben por dónde comenzar.

De manera adicional, el cambio de un paradigma de programación a otro no resulta ser un proceso sencillo de asimilar para la mayoría de desarrolladores, ya que ellos continúan implementando las buenas prácticas que utilizaron anteriormente, y buscan trasladarlas al nuevo paradigma sin considerar las dificultades técnicas que esto supone, es por ello que algunos desarrolladores se muestran reacios al cambio, de acuerdo con lo reportado por Zuhud et al. [20]. En retrospectiva algo similar ocurrió al cambiar del paradigma de programación estructurada hacia la programación orientada a objetos, de acuerdo con White et al. [21], en su momento fue un cambio radical a la manera en que se generaba el software, al implementar mecanismos de reusabilidad de código, abstracción y encapsulamiento de los datos, polimorfismo, herencia, etc.

Con el tiempo y el uso de distintas herramientas los desarrolladores finalmente aceptaron el cambio a tal punto que en la actualidad la programación orientada a objetos es la base de la mayoría del software que se construye.

Las buenas prácticas de las que se tiene registro en programación orientada a objetos no se adaptan completamente para el desarrollo de una aplicación, algunas de ellas deben ser refinadas y el cambio de paradigma de programación favorece a la aportación de una propuesta de mejores prácticas para el desarrollo de aplicaciones nativas en Android y su publicación en la plataforma de Google Play Store.



## 4. Mejores prácticas primer acercamiento

Bajo un enfoque de desarrollo empírico, se construyó una aplicación en Android. Durante el proceso y consultando la información disponible en la web de desarrolladores de Google, fue posible identificar un conjunto no exhaustivo de mejores prácticas que podrían ser de utilidad para desarrolladores con conocimientos básicos sobre Android.

Este conjunto de mejores prácticas en su primer acercamiento fue utilizado para generar diferentes aplicaciones de salud. Después de analizar los resultados obtenidos se retroalimentó el proceso y se identifica la importancia de utilizar mejores prácticas en el proceso de creación de aplicaciones en Android.

### 4.1 Mejores prácticas identificadas

Antes de presentar las mejores prácticas, de acuerdo con Jackson [22], es importante mencionar que el sistema operativo Android fue diseñado para proveer la mayor modularidad posible a los desarrolladores, lo que permite identificar de manera clara los componentes que conforman una aplicación: 1) Actividades, manejan la interfaz del usuario que se muestra en la pantalla del dispositivo móvil; 2) Servicios, procesos que se ejecutan en paralelo a la aplicación; 3) Receptores de transmisiones, se encargan de la comunicación dentro de la misma aplicación, como también de comunicarse con otras aplicaciones; y 4) Proveedores de contenido, se encargan del manejo de los datos y la administración de operaciones relacionadas con bases de datos.

Las mejores prácticas para el desarrollo de aplicaciones nativas fueron elaboradas para utilizarse en el componente de Actividades, el cual es vital para una aplicación ya que por medio de estas se presenta la interfaz gráfica al usuario a través de distintos tipos de archivos de diseño, que utilizan eXtensible Markup Language (XML).

Esta primera versión de mejores prácticas que se presenta y la cual fue reportada en la literatura por Veloz et al. [23], Impacta directamente en casi todas las fases del ciclo de desarrollo de una aplicación, se busca que con su uso y adopción dentro de los desarrolladores, se reduzca el tiempo invertido en analizar la documentación oficial, pues con base en un análisis de los requerimientos y el alcance que se pretende lograr con la

aplicación a generar, el desarrollador ya solo se referirá a la documentación oficial buscando información muy precisa para ciertos componentes.

**Tabla 2.**Primera versión de mejores prácticas

<b>Fase del ciclo de desarrollo</b>	<b>Práctica propuesta</b>	<b>Descripción</b>
<b>Requerimientos</b>	Selección de los dispositivos móviles	Analizar los requerimientos y el alcance de la aplicación a construir, y así seleccionar el dispositivo móvil que mejor se adapte a estas necesidades (tablet o smartphone).
<b>Diseño</b>	Identificar las resoluciones de pantalla a soportar	Es indispensable aprovechar al máximo el espacio disponible en cada resolución de pantalla, por lo que es conveniente realizar un estudio de mercado y determinar las resoluciones de pantalla más populares de acuerdo al dispositivo móvil seleccionado.
<b>Desarrollo</b>	Elegir un tipo de archivo de diseño	Existen diferentes tipos de archivos de diseño en la API de Android, cada uno con propiedades y comportamientos únicos, por lo que es conveniente analizar cada uno de ellos y elegir aquel que mejor se adapta a los requerimientos de la aplicación a diseñar.
<b>Liberación</b>	Llenado de la ficha de Google Play	Determinar si el producto es lo suficientemente maduro como para ser liberado en producción, alpha testing o beta testing, según corresponda. También llenar la información solicitada en cada apartado de la ficha de Google Play, ya que en base a su contenido el usuario decide descargar la aplicación.

La información que se presenta en la Tabla 2, es una síntesis de las mejores prácticas que se identificaron al realizar una aplicación nativa en Android de manera empírica, es decir, sin ser un experto en este lenguaje de programación, con conocimientos sobre el paradigma de POO, con la información de la web de desarrolladores de Google disponi-

ble y sin olvidar las etapas de un proceso de desarrollo convencional en ingeniería de software, estas prácticas son básicas en la construcción de cualquier aplicación nativa y deben ser consideradas por el desarrollador buscando reducir el tiempo de consulta de la información oficial, pensando en la portabilidad del producto final y en la integración de los diversos componentes dentro de la API de Android.

## **4.2 Evaluación de la primera versión de mejores prácticas identificadas**

En esta sección se describen los detalles y la metodología utilizada para evaluar la primera versión del trabajo propuesto. Se presentan los objetivos, las aplicaciones que se desarrollaron y una evaluación cualitativa de las mejores prácticas utilizadas.

### **4.2.1 Objetivos**

El objetivo de este caso de estudio fue preparar diferentes grupos de estudiantes de ingeniería para desarrollar aplicaciones en Android. Los estudiantes adquirieron experiencia en la construcción de aplicaciones nativas en Android para el área de salud, desempeñaron diferentes roles en su equipo, y utilizaron las mejores prácticas presentadas en la sección anterior para llevarlo a cabo. Finalmente, se aplicaron instrumentos de evaluación a los estudiantes para medir cualitativamente la adopción, claridad, ambigüedad y el uso de las mejores prácticas. De la misma manera, a los usuarios finales se les aplicó un cuestionario de usabilidad para medir la usabilidad de las interfaces de aplicaciones de salud presentadas en la sección 1.4.

### **4.2.2 Selección de los participantes**

El caso de estudio involucró 10 estudiantes de ingeniería, provenientes de distintos estados de la República Mexicana, los cuales realizaron una estancia de investigación de seis semanas en la Universidad Autónoma de Aguascalientes dentro del XIX Verano de la Investigación Científica y Tecnológica del Pacífico 2014.

### **4.2.3 Equipos**

Los estudiantes fueron organizados en pares, formando así cinco equipos donde al menos uno de sus integrantes era estudiante de ingeniería en sistemas computacionales. A cada equipo se le proporcionó: el SDK de Eclipse, una sesión informativa sobre las aplicaciones nativas y las características de las aplicaciones de e-health, los requerimientos que las aplicaciones a desarrollar y las mejores prácticas a utilizar.

#### 4.2.4 Aplicaciones

A los equipos se les permitió escoger una problemática de salud de cinco opciones disponibles. Las siguientes aplicaciones fueron desarrolladas durante el caso de estudio, después de analizar los requerimientos de cada una de ellas y compararse contra aplicaciones similares disponibles en Google Play para encontrar sus fortalezas.

Diabetes (<https://play.google.com/store/apps/details?id=test.diabetes>), para la cual se utilizó un cuestionario de 8 preguntas, conocido como cuestionario Findrisk. Dicho instrumento, de acuerdo con Bergmann et al. [24], brinda como respuesta el nivel de riesgo en que se encuentra el usuario de adquirir Diabetes Mellitus tipo II en los próximos 10 años. Adicional a esta valoración inicial se brindan consejos, sugerencias e información relevante para usuarios que padecen dicha enfermedad. Fue posible reportar los resultados parciales de esta aplicación, por Ordaz et al. [25].

Obesidad (<https://play.google.com/store/apps/details?id=com.monitoreodeobesidad>), donde se implementó un cuestionario de hábitos alimenticios el cual permite conocer los hábitos alimentarios del paciente, dichas preguntas están clasificadas en tres grupos que son: 1) Hábitos alimenticios, 2) Actividad física y 3) Estado de ánimo. Un aspecto considerado para la elección de este cuestionario fue, de acuerdo con Rodríguez et al. [26], por medio de las preguntas realizadas se obtiene información sobre el consumo de azúcar en la dieta, incluso sobre el tipo de alimentos que frecuentemente consumen los usuarios con obesidad. Además, el cuestionario empleado, permite identificar de manera más precisa el estado de salud de un usuario ya que considera factores como el consumo de azúcar, alimentación saludable, ejercicio físico, contenido calórico, bienestar psicológico, tipo de alimentos, conocimiento y control y por último consumo de alcohol. Esta información junto con el cálculo energético basal proporciona una valoración inicial del estado de salud del usuario, también de esta aplicación se logró publicar un artículo por Mariscal et al. [27].

Drogadicción (<https://play.google.com/store/apps/details?id=test.drogadiccion>), implementa un cuestionario de 24 preguntas diseñado por la Universidad de Rhode Island, el laboratorio HABITS (Health and Addictive Behaviors: Investigating Transtheoretical Solutions). Dicho instrumento de acuerdo con Prochaska & DiClemente [28], proporciona un diagnóstico con el nivel de toxicomanía o disposición al cambio del indivi-

duo. El prototipo fue publicado en Google Play e incluso fue evaluado por el Instituto Aguascalentense de la Juventud para ser utilizado en las valoraciones iniciales de los adolescentes.

Ingesta de alcohol (<https://play.google.com/store/apps/details?id=test.alcoholismo>) la cual implementa el cuestionario AUDIT. De acuerdo con Saunders et al. [29], este cuestionario permite identificar los trastornos debidos al consumo de alcohol. Las aplicaciones existentes que tratan temas relacionados con la ingesta de alcohol, no cuentan con el nivel de madurez suficiente para diagnosticar oportunamente los trastornos ocasionados por el consumo en exceso de esta sustancia.

Depresión disponible en (<https://play.google.com/store/apps/details?id=test.depresion>), los resultados de esta aplicación se basan en la implementación del algoritmo CES-DR creado por especialistas. Como presenta González-Forteza et al. [30], este instrumento tiene excelentes características psicométricas en adolescentes mexicanos, por lo que es adecuada para la detección oportuna de depresión. Esto permite utilizarla en las vigilancias sistemáticas y periódicas en la población escolar para detectar necesidades de atención, prevención y promoción de la salud mental.

#### 4.2.5 Instrumentos de evaluación

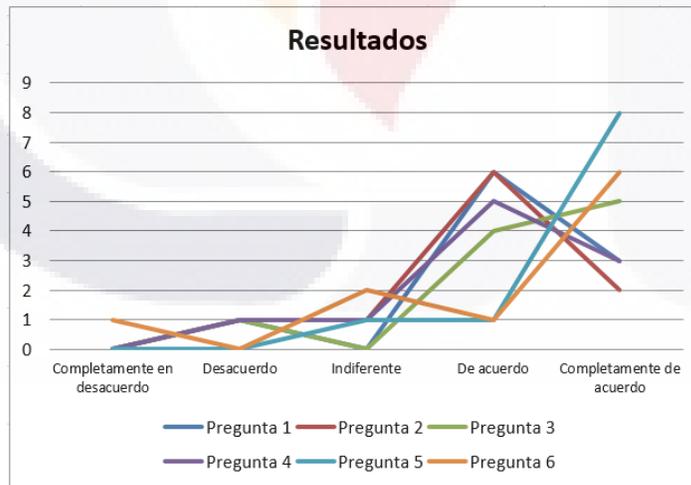
Se elaboró un primer instrumento de evaluación buscando medir que tan adecuadas resultaron las mejores prácticas presentadas con las características solicitadas al inicio, su claridad, su uso y adopción entre los desarrolladores. El instrumento de evaluación se muestra en la Gráfica 4.

**Instrucciones:** Por favor, lea cuidadosamente cada una de las afirmaciones presentadas a continuación y, (de acuerdo a su percepción) seleccione el grado de cumplimiento para cada una de ellas.

Con respecto a las mejores prácticas presentadas en el curso:	Fuertemente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
1. Son adecuadas con las características del producto que se quiere desarrollar.					
2. Son claras, describen lo que se debe hacer sin ninguna ambigüedad.					
3. Su uso permite generar aplicaciones nativas Android de calidad.					
4. Su uso reduce el tiempo de desarrollo.					
5. Resultaron de utilidad para el desarrollo de las aplicaciones de e-health					
6. Las utilizaría posteriormente para el desarrollo de proyectos o aplicaciones en Android.					

**Gráfica 4.** Instrumento de evaluación

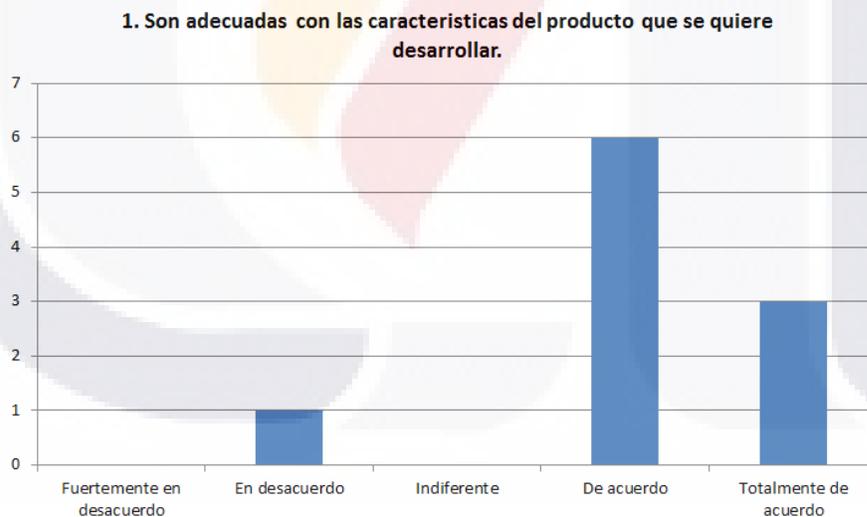
Las primeras tres sentencias se formularon en base al capítulo 8 Garantía de calidad del software (SQA/GCS) del libro Ingeniería de software, un enfoque práctico, 5ta edición, Mc Graw Hill. Mientras que las últimas tres sentencias se plantearon en base al capítulo 27, Gestión de la calidad del libro Ingeniería de software, 7ma edición, Editorial Pearson. Los resultados obtenidos de la aplicación del instrumento se muestran en la gráfica de la Gráfica 5.



**Gráfica 5.** Resultados del instrumento de evaluación

De las respuestas obtenidas para cada pregunta se tienen las siguientes interpretaciones:

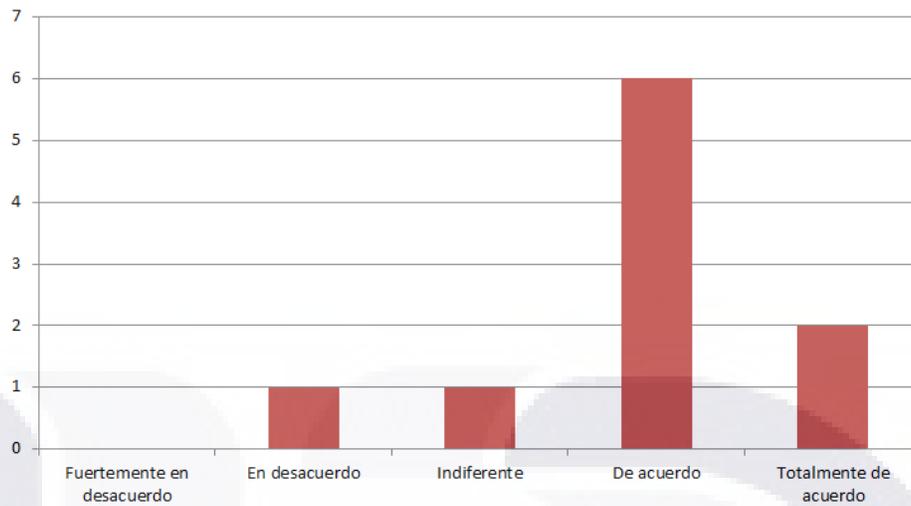
- Pregunta 1, la mayoría de los estudiantes (nueve) consideran que las mejores prácticas se adecuan a las características mencionadas para aplicaciones de e-health.
- Pregunta 2, ocho estudiantes consideran que las mejores prácticas presentadas, son claras y describen qué se debe hacer.
- Pregunta 3, nueve estudiantes consideran que el uso de las mejores prácticas permite generar aplicaciones nativas de calidad.
- Pregunta 4, ocho estudiantes expresan que permiten generar aplicaciones de calidad y su uso reduce el tiempo de desarrollo.
- Pregunta 5, a la mayoría de los estudiantes les parecieron lo suficientemente útiles para desarrollar sus aplicaciones.
- Pregunta 6, la adopción no es generalizada, es decir, anteriormente se dijo que las mejores prácticas eran útiles, reducían tiempo de desarrollo y se ajustaban al producto que se deseaba crear, sin embargo, su adopción para utilizarlas en proyectos futuros no es alta, pues apenas seis de los diez estudiantes considerarían utilizarlas más adelante.



**Gráfica 6.** Pregunta uno, primera versión.

De esta primera imagen, se observa que la mayoría de los estudiantes consideran que las mejores prácticas se adecuan a las características mencionadas para aplicaciones de e-health, seis de ellos están “de acuerdo”, tres “totalmente de acuerdo” y solo uno “en desacuerdo”.

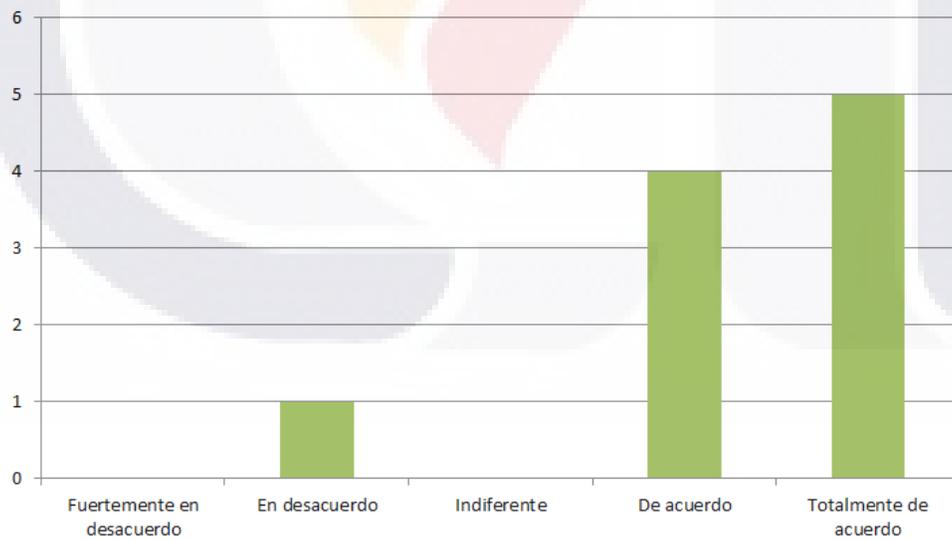
**2. Son claras, describen lo que se debe hacer sin ninguna ambigüedad.**



**Gráfica 7.** Pregunta dos, primera versión.

Si bien la mayoría de los estudiantes está de acuerdo en que las mejores prácticas presentadas son claras, no es unánime la respuesta. De la gráfica se puede observar que seis estudiantes expresan su opinión como “de acuerdo”, dos como “totalmente de acuerdo”, uno es “indiferente” y finalmente uno “en desacuerdo”.

**3. Su uso permite generar aplicaciones nativas Android de calidad.**

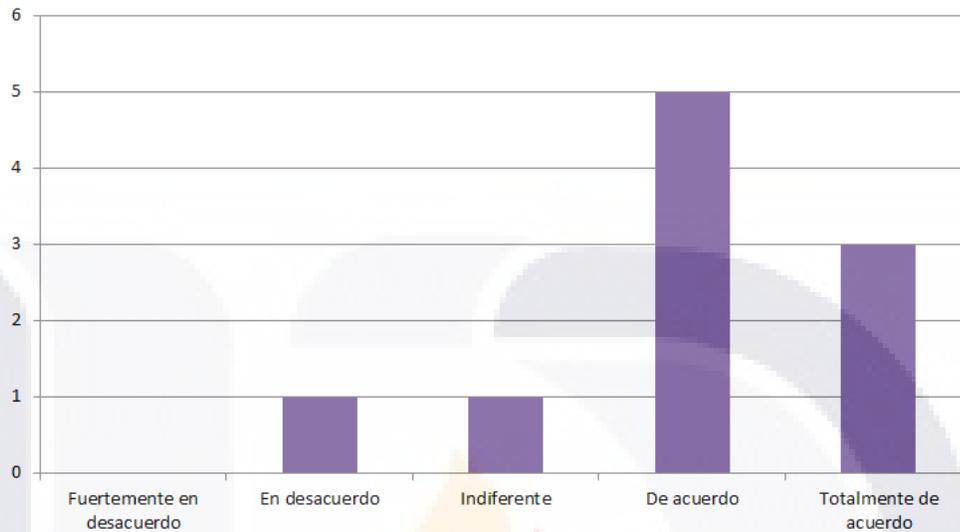


**Gráfica 8.** Pregunta tres, primera versión.

La mayoría de los estudiantes consideran que su uso permite generar aplicaciones de calidad conforme a las características que les fueron presentadas al inicio. Cinco estu-

diantes están "totalmente de acuerdo", cuatro "de acuerdo" y solo uno "en desacuerdo".

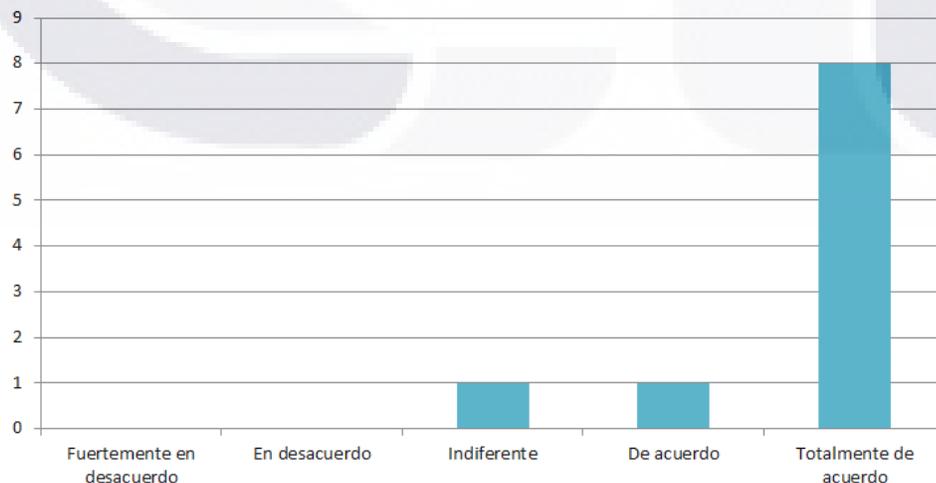
**4. Su uso reduce el tiempo de desarrollo.**



**Gráfica 9.** Pregunta cuatro, primera versión.

De la Gráfica anterior se puede observar que ocho de los diez estudiantes están de acuerdo en que el trabajo presentado es lo suficientemente útil como para reducir el tiempo en el desarrollo, cinco de ellos contestaron "de acuerdo", tres "totalmente de acuerdo", uno "indiferente" y finalmente uno "en desacuerdo".

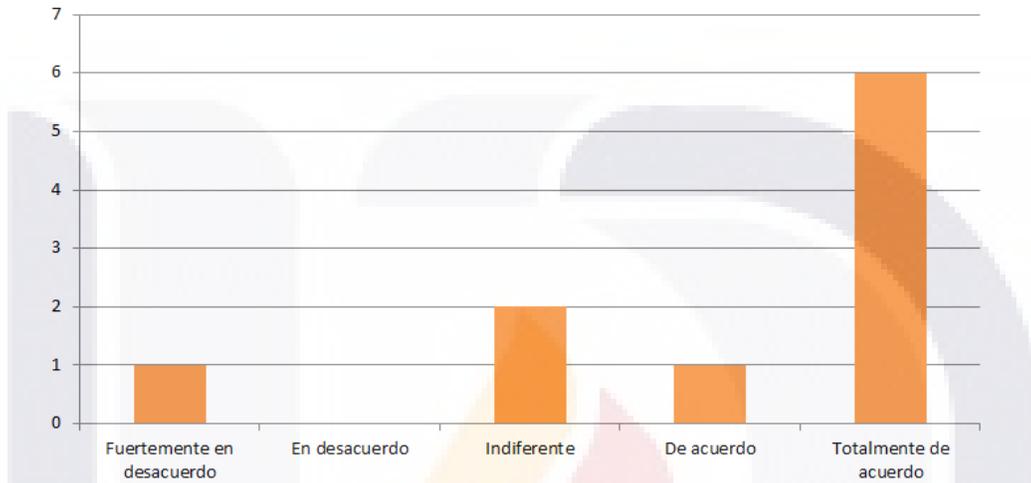
**5. Resultaron de utilidad para el desarrollo de las aplicaciones de e-health**



**Gráfica 10.** Pregunta cinco, primera versión.

A la mayoría de los estudiantes les parecieron lo suficientemente útiles para desarrollar sus aplicaciones de acuerdo con los requerimientos y características presentadas al inicio del curso. Ocho de los estudiantes respondieron “totalmente de acuerdo”, uno “de acuerdo” y uno más como “indiferente”.

**6. Las utilizaría posteriormente para el desarrollo de proyectos o aplicaciones en Android.**



**Gráfica 11.** Pregunta seis, primera versión.

En cuanto a esta última pregunta, la adopción no es generalizada, es decir, anteriormente se dijo que las mejores prácticas eran útiles, reducían tiempo de desarrollo y se ajustaban a las características del producto que se deseaba crear, sin embargo, su adopción para utilizarlas en proyectos futuros no es alta, pues apenas seis de los diez estudiantes respondieron “totalmente de acuerdo”, uno “de acuerdo”, dos “indiferente” y uno más “fuertemente en desacuerdo”.

Para comparar los resultados del primer instrumento de evaluación, se diseñó otro más, para el cual se tomó como base el estándar de Moprosoft y determinando que actividades de este modelo se ajustaban más con las características de las mejores prácticas propuestas. Por ejemplo, las actividades orientadas a las pruebas e integración de componentes, por el momento no se relacionan con el trabajo presentado.

Este segundo instrumento, se aplicó a ocho de los diez estudiantes apenas una semana después de haberse aplicado el primer instrumento, es decir, en el transcurso de la quinta semana de su estancia de investigación. La razón de porque se aplicó solo a

ocho personas, es que todos los equipos a excepción de uno ya contaban con un producto listo para publicarse. A estos equipos se les mostró la segunda parte de las mejores prácticas propuestas, que están relacionadas con la publicación de una aplicación en Google Play, las cuales son:

- *Llenar la ficha de Google Play, llenar correctamente la ficha de Google Play con los textos e imágenes que solicita, brinda un panorama general del contenido de la aplicación. Seleccionar los países e idiomas en los cuales estará disponible la aplicación para descargarse, así como el precio y selección de la clase de contenido. Con esta información el usuario final decide si descarga o no la aplicación.*
- *Liberar la aplicación, identificar si el producto es lo suficientemente maduro como para publicarse directo en producción, o bien, aún se encuentra en una etapa temprana y es recomendable liberarle en alpha o beta testing, para identificar y solucionar errores. Atender posteriormente las críticas y errores recibidos, para generar una nueva versión que corrija las fallas detectadas.*

En la Tabla 3, se presentan los resultados obtenidos de haber aplicado este segundo instrumento con los equipos que liberaron sus aplicaciones en Google Play.

**Tabla 3.**Resultados del segundo instrumento de evaluación.

A1. Realización de la fase de inicio	SI	NO	Indiferente
Fueron claras las características de esta familia de productos.	8	0	0
A2. Realización de la fase de requisitos			
Los requisitos de la aplicación a desarrollar/modificar permitieron determinar el alcance y factibilidad del producto a construir.	7	0	1
A3. Realización de la fase de análisis y diseño			
El análisis de los requerimientos, permite describir la estructura básica de la aplicación y sus componentes.	8	0	0
Los requerimientos permiten describir la apariencia y el comportamiento de la interfaz de usuario.	7	1	0
A4. Realización de la fase de construcción			
Construcción/modificación de los componentes de software con base en el Análisis y Diseño.	8	0	0

Uso de los dispositivos virtuales/físicos para probar el funcionamiento de los componentes construidos.	8	0	0
Corrección de los defectos encontrados hasta lograr pruebas sin defectos.	5	2	1
<b>A5. Liberación del producto terminado</b>			
Es claro cómo generar el archivo apk para liberar la aplicación.	8	0	0
Se describe sin ninguna ambigüedad como llenar la ficha de Google Play.	8	0	0
Describen la importancia de atender las recomendaciones/críticas de los usuarios para mejorar el producto publicado.	8	0	0

Con los resultados de estos dos instrumentos de evaluación fue posible hacer correcciones sobre las mejores prácticas que se presentaron al inicio, esta retroalimentación permitió generar una segunda versión de mejores prácticas, la cual se presenta en la siguiente sección.

### 4.3 Mejores prácticas segunda versión

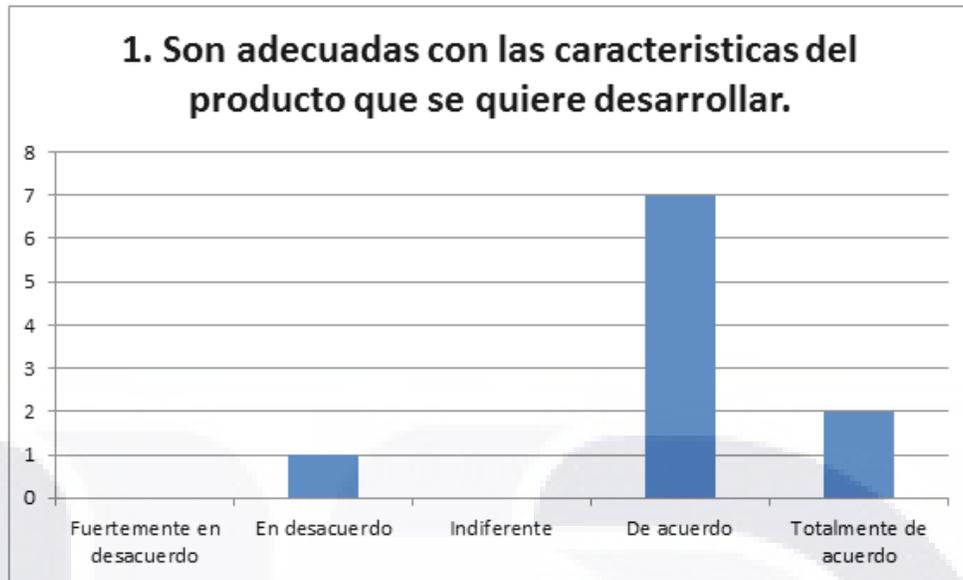
Se generó una segunda versión de las mejores prácticas tomando en cuenta los resultados de los instrumentos de evaluación utilizados, y consultando de nuevo la documentación oficial de Android [31]. Se eliminó una práctica ya que generaba confusión entre los estudiantes pues no quedaba clara su utilidad. Esta segunda versión del trabajo propuesto fue utilizada por los estudiantes para realizar los cambios en sus aplicaciones en base a los comentarios recibidos a través de Google Play.

**Tabla 4.** Segunda versión de mejores prácticas.

Fase	Práctica propuesta	Descripción
<b>Requerimientos</b>	Elegir los dispositivos donde correrá la aplicación	Es importante determinar si la aplicación se ejecutara en smartphones, tablets, o ambos. El tamaño de pantalla de estos dispositivos puede variar considerablemente, una Tablet, permite incorporar más elementos a la interfaz de usuario, al contrario de un smartphone que no cuenta con ese espacio disponible.
<b>Diseño</b>	Identificar las resoluciones de	Revisar las resoluciones de panta-

	pantalla con las que se trabajara	lla más populares en el mercado para el tipo de dispositivo seleccionado. E.g. Smartphone 5.7 y Tablet 7.0 y 10.1 pulgadas.
<b>Desarrollo</b>	Selección de los archivos de diseño	La selección de los archivos de diseño, determina el comportamiento que tendrá la interfaz de usuario y como se organizaran los elementos dentro de la pantalla. Existen diferentes opciones disponibles dentro de la API de Android, por lo que es conveniente revisar la información disponible.
<b>Liberación</b>	Completar la ficha de Google Play	La ficha de Google Play brinda información técnica y una descripción del propósito de la aplicación, sirve a los usuarios como guía antes de descargar. Cuenta con diferentes mecanismos que permiten la comercialización de las aplicaciones en aquellos países donde Google tiene presencia. La opinión de los usuarios propicia la mejora continua del producto.

Al final de la estancia de investigación, se aplicó a los estudiantes el primer instrumento de evaluación agregando la pregunta ¿se podría haber generado el mismo resultado sin ayuda de las mejores prácticas definidas?, con esto se buscó medir si hubo una mejoría en cuanto al producto presentado, teniendo los siguientes resultados.



**Gráfica 12.** Pregunta uno, segunda versión.

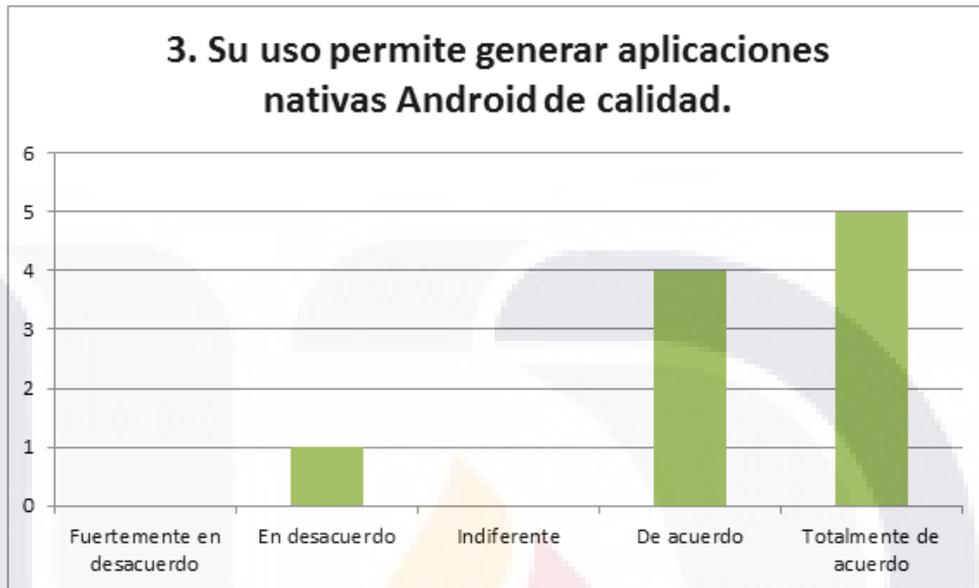
En la Gráfica 12 se puede observar que a diferencia de la primera ocasión que se hizo esta pregunta, una persona cambió de opinión, de totalmente de acuerdo a de acuerdo. Sin embargo, la mayoría (nueve) consideran que si son aptas para las características del producto que se pidió desarrollar (aplicaciones de e-health).



**Gráfica 13.** Pregunta dos, segunda versión.

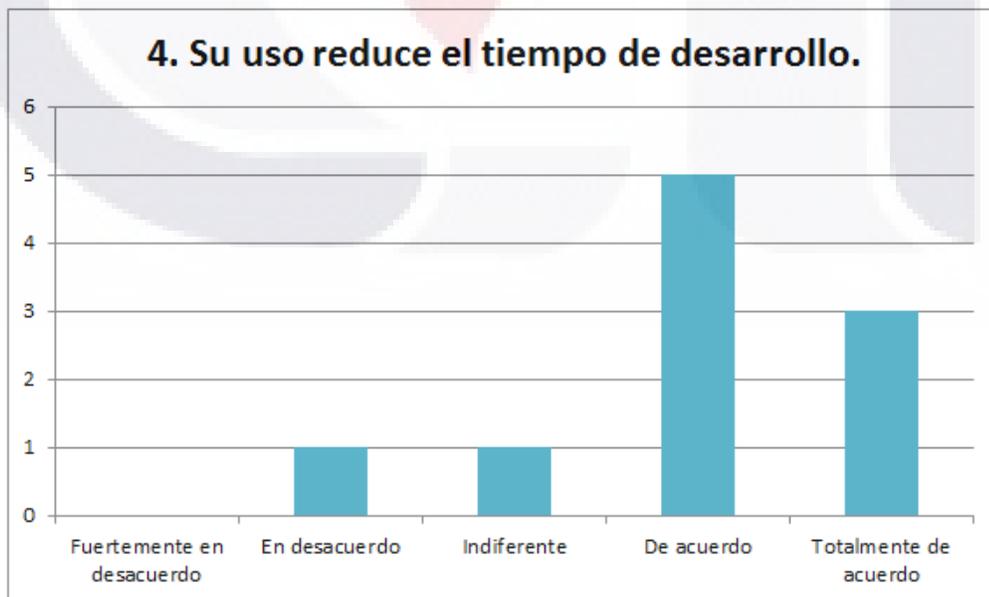
Esta nueva versión de las mejores prácticas parece ser más clara que en comparación con la versión anterior, pues nueve de los estudiantes las consideran así. En la primera

aplicación del instrumento se tenían seis personas en “de acuerdo” y dos en “totalmente de acuerdo”, en esta ocasión se tienen cuatro en “de acuerdo” y cinco en “totalmente de acuerdo”.



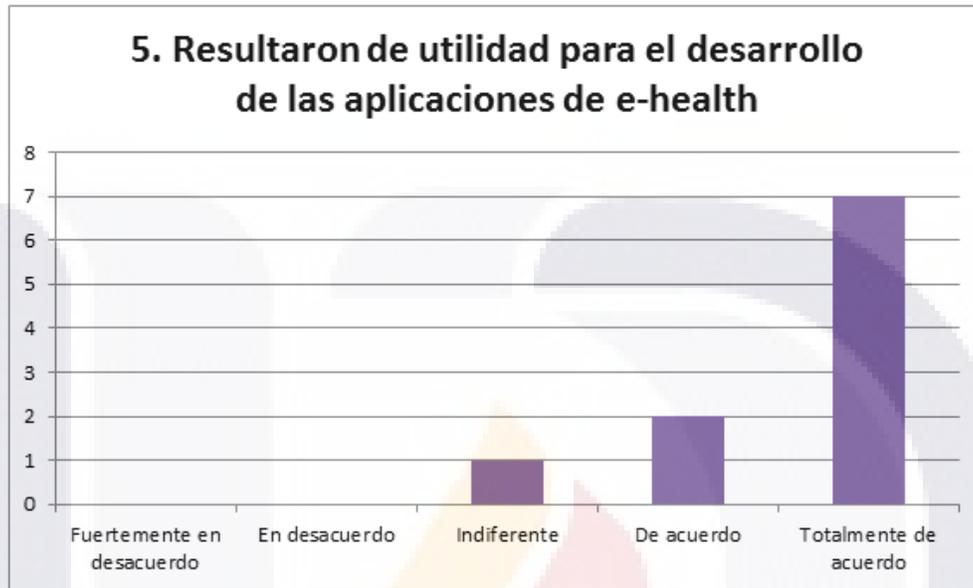
**Gráfica 14.** Pregunta tres, segunda versión.

En cuanto a la pregunta tres, no cambian de los resultados obtenidos en comparación con la primera aplicación de este instrumento. Los estudiantes consideran que esta segunda versión del trabajo propuesto les permite crear aplicaciones nativas de calidad.



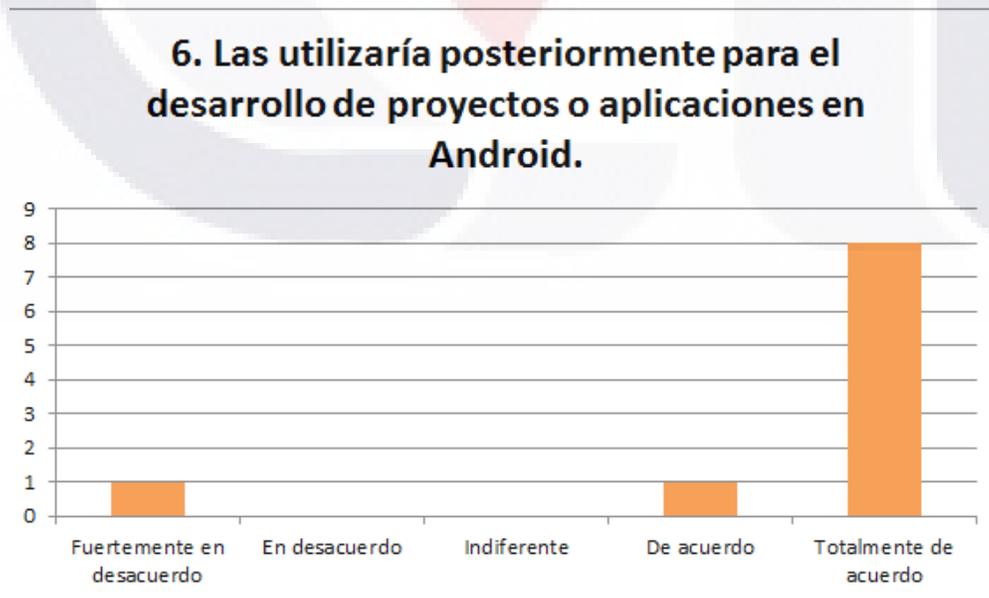
**Gráfica 15.** Pregunta cuatro, segunda versión.

La pregunta cuatro, los resultados son los mismos que en la ocasión anterior, no hubo cambio alguno. Es decir, esta nueva versión de las mejores prácticas permite generar aplicaciones de calidad y su uso reduce el tiempo de desarrollo, de acuerdo con las gráficas de las Gráficas 14 y 15 respectivamente.



**Gráfica 16.** Pregunta cinco, segunda versión.

De los resultados obtenidos para esta segunda versión, cabe señalar que una persona cambió de opinión, de "totalmente de acuerdo" a "de acuerdo".



**Gráfica 17.** Pregunta seis, segunda versión.

El cambio más significativo es que en comparación con la versión anterior, la adopción de las mejores prácticas es prácticamente generalizada, ya que nueve estudiantes las utilizarían en futuros proyectos de desarrollo en Android, ocho estudiantes respondieron con "totalmente de acuerdo", uno con "de acuerdo" y otro con "fuertemente en desacuerdo".



**Gráfica 18.** Pregunta siete, segunda versión.

Finalmente, con esta pregunta no se buscó medir las cualidades de los estudiantes, lo que se pretendía conocer era, si en caso de no haberseles mostrado o explicado el trabajo de mejores prácticas el resultado hubiera sido el mismo. Es decir, sin ningún de los materiales de apoyo utilizados, ¿habrían logrado el mismo resultado, en el mismo tiempo y con una calidad similar? Vemos que 6 de los 10 estudiantes consideran que no habría sido así.

#### 4.4 Productos generados

Los resultados parciales del verano de investigación permitieron avanzar hacia una nueva versión de las mejores prácticas para el desarrollo de aplicaciones nativas en Android, así como la publicación en Google Play Store de cinco aplicaciones. A continuación se presentan los resultados obtenidos de manera cualitativa que son comentarios y valoraciones de los usuarios junto con el número de descargas conseguidas hasta la fecha.

En las Gráficas 19, 20 y 21 se presentan algunas pantallas de las aplicaciones cuestionario sobre depresión, diabetes II (DM II) y test de alcoholismo, en las cuales se puede observar que siguen las guías de diseño que fueron presentadas en el capítulo 2, lo que hace que estas aplicaciones sean una familia de productos donde siguen una línea definida de diseño y desarrollo.



**Gráfica 19.** Pantallas de la aplicación Cuestionario sobre depresión

En la Gráfica 19 se pueden observar el menú principal, una pregunta del cuestionario sobre depresión y la pantalla de créditos. Contienen los elementos y características que las interfaces de aplicaciones de e-health deben contener como los logotipos, la distribución de los elementos y el correcto uso de navegación por mencionar algunos ejemplos.

La aplicación Diabetes II o DM II utiliza un cuestionario de ocho preguntas proveniente de una junta de Asociaciones de Diabetes el cual brinda como respuesta el nivel de riesgo en el que te encuentras de adquirir Diabetes Mellitus Tipo II en los próximos 10 años, además ofrece una serie de consejos alimenticios como físicos para una vida más saludable.



**Gráfica 20.** Pantallas de la aplicación Diabetes II

En la Gráfica 20 se muestran las capturas de la pantalla de bienvenida al igual que el modulo donde se calculó el índice de masa corporal del usuario, por lo cual se requiere que este ingrese su altura y peso para continuar.



**Gráfica 21.** Pantallas de la aplicación Test alcoholismo

En la Gráfica 21 se observan el menú principal, una pregunta del cuestionario al igual que la pantalla de créditos de la aplicación test alcoholismo. De manera similar a otras aplicaciones presentadas anteriormente hace uso de logotipos, la distribución de los elementos y el correcto uso de la navegación.

En la Gráfica 22 se presentan los datos estadísticos obtenidos de la consola para desarrolladores de Google, la cual permite conocer el número de descargas acumuladas, comentarios, valoraciones de los usuarios, entre otras características que permiten dar un seguimiento al producto una vez que este ha sido publicado.

NOMBRE DE LA APLICACIÓN	PRECIO	INSTALACIONES ACTUALES/TOTALES	VALORACIÓN MEDIA / TOTAL	ERRORES Y ANRS	ÚLTIMA ACTUALIZACIÓN	ESTADO
 Cuestionario sobre Depresión 2.1	Gratuita	81 / 1.043	★ 4,29 / 24	—	25/7/2014	Publicada
 Diabetes II 1.3	Gratuita	40 / 392	★ 4,44 / 16	1	24/7/2014	Publicada
 Monitoreo de Obesidad 1.2	Gratuita	78 / 915	★ 4,13 / 8	—	9/2/2014	Publicada
 Test Alcoholismo 2.0	Gratuita	25 / 248	★ 4,67 / 9	—	22/7/2014	Publicada
 Test de Obesidad 1.0	Gratuita	19 / 226	★ 3,20 / 5	2	28/7/2014	Publicada
 Test Dependencia a las Drogas 1.0	Gratuita	4 / 48	★ 3,75 / 4	—	22/7/2014	Publicada
 Test sobre Depresión 1.2	Gratuita	488 / 7.751	★ 3,61 / 62	—	7/1/2014	Publicada

**Gráfica 22.** Resultados de las aplicaciones publicadas

La aplicación Cuestionario sobre Depresión cuenta con más de mil descargas una valoración promedio de 4.29 estrellas de un máximo de 5 y ha recibido comentarios como "Muy bueno, lo recomiendo", "Creo que es muy específico y clara, a lo que se refiere en las preguntas y significado del vocabulario.", "Me encantó". Esta aplicación ha sido descargada principalmente en México, España y Argentina debido a que no se limitó su disponibilidad en el mercado.

La aplicación Diabetes II cuenta con 400 descargas y una valoración promedio de 4.44 estrellas. Se han recibido comentarios como "Interesante", "Está muy buena, fácil de usar y medicamente muy útil", "Me parece una buena aplicación se las recomiendo a todos...te dice como mantener tu salud al 100% y para no correr riesgo de diabetes". De manera similar, los países donde se han obtenido un mayor número de descargas son México, España y Brasil.

La aplicación Monitoreo de Obesidad ha generado más de 900 descargas hasta la fecha y una valoración media de 4.13 estrellas. Descargada principalmente en México, España y Argentina.

Además de las prácticas presentadas para el diseño de la aplicación se utilizó también la misma estrategia para la publicación en Google Play. Como se indicó en la Tabla 4 cada una de las aplicaciones muestra el icono para descargar, una imagen general que representa el concepto o temática principal del producto, una descripción breve donde se indican los instrumentos realizados, el tipo de diagnósticos, información y/o consejos que obtendrá el usuario así como los países y el precio para descargar las aplicaciones.

Es por esto que los resultados obtenidos hasta el momento no varían mucho uno de otro, reflejan la percepción del usuario sobre el producto final.

Cabe mencionar que es necesario conducir un estudio de usabilidad con el objetivo de medir el impacto de las mejores prácticas, y las plantillas para el diseño de interfaces de usuario de aplicaciones de salud. Esto permitirá identificar las áreas de oportunidad del trabajo propuesto, buscando con ello mejorar la experiencia del usuario al ofrecerle un producto integral específicamente diseñado para brindar información médica con una base científica válida.



## 5. Herramientas para el diseño de aplicaciones nativas

Existen diferentes herramientas disponibles para el desarrollo de aplicaciones nativas en Android. Cada una de ellas tiene sus ventajas y desventajas. Cabe mencionar que no son excluyentes, es posible utilizar diferentes enfoques para el desarrollo de software móvil. Por lo que al conocer el impacto de cada herramienta en conjunto con una estrategia y alcances definidos, es posible generar aplicaciones para el sistema operativo Android.

### 5.1 Modelos para el desarrollo de software

Como se mencionó anteriormente las aplicaciones para dispositivos móviles son diferentes de los programas que se desarrollan para cómputo tradicional. Las diferencias van desde los ambientes de desarrollo y herramientas utilizadas para su construcción, los recursos disponibles al momento de su ejecución, la interacción que tienen estos productos con el usuario final, pero sobretodo el corto tiempo para su desarrollo y liberación disponible en comparación con un sistema para PCs de escritorio.

Las aplicaciones para dispositivos móviles pueden ser muy complejos en su diseño y construcción. En el capítulo 3 se identifican algunos aspectos presentes al momento de aprender los nuevos elementos dentro de un API y su interacción. Por otra parte Brooks [32], identifica dos tipos de complejidad en su libro *The Mythical Man-Month*, los cuales son: complejidad esencial, la cual es inherente e inseparable del problema, como la complejidad presente en cualquier problema NP la cual no puede ser eliminada. Y la complejidad accidental, la cual es una consecuencia directa de las estrategias utilizadas para resolver dicho problema.

Existen distintos enfoques que buscan reducir el grado de la complejidad accidental uno de ellos es la ingeniería dirigida en modelos (MDE por sus siglas en ingles). Dentro de MDE el modelo se considera como un artefacto clave, el cual es utilizado en todas las fases de la ingeniería de software (análisis, diseño, implementación y pruebas). El objetivo principal de este enfoque es reducir la brecha que existe entre el dominio del problema y la implementación de la solución. De acuerdo con Parada & Brisolará [33] el uso efectivo de este paradigma, al combinarse los modelos con otras herramientas,

proporciona abstracción, automatización, y se puede reducir el tiempo de desarrollo hasta en un 70%.

Ingeniería de software puede definirse como el proceso por el cual un individuo o equipo se organiza y administra durante la creación de un sistema de software, desde su conceptualización hasta su liberación como producto final.

De acuerdo con Selic [34], un modelo de ingeniería de algún sistema, es una representación reducida de ese sistema que resalta las propiedades de interés para un punto de vista determinado. Un modelo de ingeniería útil debe contener las siguientes características:

- Abstracción, el modelo debería remover u ocultar todos los detalles irrelevantes, entonces lo esencial sobresale.
- Entendible, el modelo debería ser expresado de forma que transmita la información esencial de una manera directa y precisa.
- Preciso, el modelo debe reflejar correctamente las propiedades de interés del sistema modelado.
- Predictivo, el modelo debe ser capaz de predecir correctamente el comportamiento y otras propiedades del sistema modelado.

El propósito de MDE es que el modelo pueda ser utilizado en las distintas fases del ciclo de desarrollo, de manera que al conjuntar su uso con otras herramientas se pueda dar un seguimiento adecuado de las entradas y salidas a través de cada etapa del modelo.

La combinación de MDE para el desarrollo de software junto con herramientas y/o técnicas del área de HCI (Human Computer Interaction), para diseñar productos centrados en el usuario constituye una alternativa para desarrollar aplicaciones nativas en Android. Con un modelo basado en mejores prácticas que guíe a los usuarios durante las etapas del ciclo de vida del producto, independiente de la plataforma o ambientes de desarrollo, incluyendo técnicas de HCI para mantener un cierto nivel de portabilidad, desempeño e interacción, permite a los desarrolladores identificar las entradas y salidas a lo largo del proceso de manera que el producto vaya progresando hasta que tenga la calidad suficiente para ser publicado.

En la medida en que las aplicaciones móviles se vuelven más complejas, pasando de ser productos para fines recreativos a ser productos que administran información sensible del usuario, será necesario aplicar procesos de ingeniería de software para asegurar que las aplicaciones son seguras, portables y de calidad por mencionar algunos factores críticos. Algunas técnicas de gestión y desarrollo podrán emplearse en el dominio de dispositivos móviles sin algún cambio, mientras que algunas otras tendrán que adaptarse o replantearse para ajustarse a las necesidades que demanda este paradigma.

## 5.2 Ambientes de desarrollo

La web de desarrolladores de Android provee un Software Development Kit (SD) y un conjunto de herramientas como un apoyo a los desarrolladores de aplicaciones móviles. Desde sus inicios hasta el momento de escribir esta tesis, el SDK oficial era Eclipse que junto con un módulo conocido como ADT (Android Development Tools) constituían una opción para desarrollar aplicaciones basadas en lenguaje Java. Ambas herramientas en conjunto proveían características como acceso a la interfaz gráfica de usuario y sus elementos, uso de la ventana de comandos, herramientas para el diseño de interfaces de usuario para un rápido prototipo, diseño, y construcción de la interfaz gráfica de la aplicación.

Sin embargo, desde el momento en que se presentó al público la nueva versión del sistema operativo Android, 5.0 Lollipop, el nuevo SDK oficial es Android Studio. A través de la misma web de desarrolladores se presentan manuales que permiten a los desarrolladores migrar sus proyectos de Eclipse a Android Studio. Este nuevo SDK ha mejorado su editor de código, ahora es capaz de completar las instrucciones de las líneas de código y análisis del mismo. Se han integrado el uso de plantillas para el diseño en dispositivos que funcionen con el sistema operativo Android, por mencionar algunas de sus principales características, para mayor información sobre mejoras y requerimientos de instalación, refiérase a la siguiente dirección dentro de la web de desarrolladores <http://developer.android.com/sdk/index.html#Requirements>.

De manera similar, existen distintos ambientes de desarrollo en la web, pueden contar con soporte multiplataforma, o bien, con soporte para un producto en específico, como

los videojuegos. Un ejemplo de esto, es los gráficos o el motor de física que ofrece Unity. Corona SDK representa otra opción multiplataforma, utiliza el lenguaje Lua, un lenguaje de scripting enfocado precisamente para desarrollar videojuegos. Netbiscuits, Appcelerator, Kyte, entre otras compañías, ofrecen herramientas y frameworks para el soporte y creación de aplicaciones móviles web a través de sus propios SDKs.

PhoneGap [35], es un framework de código abierto para la construcción de aplicaciones nativas a través de tecnologías web como HTML, CSS, Javascript, incluyendo HTML5 y CSS3, de igual manera es una herramienta multiplataforma con soporte para Android, iOS, Windows Phone, Blackberry, bada, webOS. En el caso concreto de Android, permite el uso del acelerómetro, cámara, brújula, contactos, archivos, geo localización, media, red, notificaciones (alertas, sonidos y/o vibraciones) además de permitir acceder al almacenamiento del dispositivo.

En Octubre de 2011, PhoneGap fue donado a la Apache Software Foundation (ASF) bajo el nombre de Apache Cordova. A través de la ASF, el desarrollo futuro de PhoneGap asegurara la administración abierta del proyecto. Permanecerá abierto y de código abierto bajo la licencia de Apache, Versión 2.0.

De manera similar PhoneGap [35], cuenta con una página oficial y un sitio con información para desarrolladores, donde se puede encontrar recursos para el manejo de los sensores, configuración de los archivos con extensión xml, y otras referencias del uso de la API.

Para crear una aplicación en PhoneGap es necesario crear un sitio web primero, para ello debe utilizarse HTML, CSS y/o Javascript, no se requiere del uso de alguna herramienta adicional, sin embargo, es conveniente organizar los archivos en carpetas y crear una estructura de directorios como un sitio web convencional. Ya que se tiene listo la página web, se puede sincronizar el contenido con un proyecto en GitHub, esta herramienta es para la administración de proyectos, o bien, se puede comprimir el contenido del sitio en un archivo con extensión rar. Finalmente, habrá que crear una cuenta en PhoneGap y subir el archivo comprimido, o en su lugar enlazar con el directorio en GitHub. La aplicación estará disponible en minutos para la plataforma seleccionada a

través de un código QR que se puede escanear para descargar y probar la aplicación directamente en el dispositivo.

Firtman [6], en su libro "Programming the Mobile Web" muestra que aunque una aplicación construida con PhoneGap solo puede utilizar HTML, CSS y Javascript, es posible por medio del uso de protocolos de red comunicarse con el motor detrás de la vista. Esto permite que una aplicación en PhoneGap pueda acceder de manera remota a servicios de negocios, bases de datos en otros lenguajes (e.g. PHP, ASP, JSF, Java, .Net, etc).

De acuerdo con Babu et al. [36], PhoneGap se ha vuelto popular entre los desarrolladores, por el soporte multiplataforma, el uso de tecnologías web hace que la curva de aprendizaje no sea tan larga en comparación con el manejo del lenguaje del SDK oficial, esto se traduce directamente en un menor costo del proyecto y un menor tiempo para su publicación. Otra característica importante es que permite una mayor portabilidad de la aplicación, su desempeño es comparable al de una aplicación desarrollada a través del SDK oficial y, la interfaz de usuario se beneficia del uso de CSS3 y HTML5 para ser atractiva visualmente.

### 5.3 Metodologías ágiles

Los métodos ágiles o desarrollo ágil, es un modelo de desarrollo basado en iteraciones, donde en cada iteración se realizan todas las fases del ciclo de desarrollo. En febrero del 2001, se celebró una reunión en Utah, a la cual asistieron 17 desarrolladores de software, representantes de las más populares y modernas metodologías en ese entonces; producto de esta reunión fue la publicación del Manifiesto Ágil [37]. En dicho manifiesto se plasma la filosofía y los valores a tener en cuenta en la elaboración de software utilizando las ahora llamadas metodologías ágiles. Del manifiesto ágil, resalta lo siguiente donde se valora a:

- Los individuos e interacciones sobre los procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación extensiva

- Respuesta ante el cambio sobre seguir un plan

Hay metodologías ágiles que se han hecho un lugar en el gusto de desarrolladores y empresas por igual, tal es el caso de Scrum. De acuerdo con Hossain et al. [38], el desarrollo basado en Scrum se logra mediante iteraciones con duración de dos a cuatro semanas en las cuales el software se desarrolla en incrementos llamados "sprints". Cada uno de estos sprints comienza con una planeación y termina con una revisión detallada de los planes para el sprint. Hay reuniones programadas de los diferentes actores para analizar la situación del negocio, del mercado, de la tecnología entre otros factores relevantes al producto. También hay reuniones diarias con los miembros del equipo que sirven de retroalimentación para el equipo de desarrollo. La tendencia del uso de esta metodología ha ido a la alza en los últimos años en proyectos globales de desarrollos de software (GSD por sus siglas en inglés) y se pueden adaptar estas características a equipos con un número reducido de integrantes.

La mayor parte de los principios y buenas prácticas de estas metodologías requieren de cierta experiencia para su empleo. De acuerdo con Scharff [39], es recomendable seguir los siguientes principios de las metodologías ágiles para el diseño de aplicaciones móviles:

- El cliente debe estar constantemente involucrado en el proceso de desarrollo (se puede considerar al usuario final como el cliente ya que es él quien finalmente utilizará la aplicación)
- Los requerimientos deben ser capturados en un alto nivel
- El equipo debe aceptar la realidad de los cambios en los requerimientos
- Los requerimientos deben ser priorizados utilizando la regla de 80/20, es decir, el 80% del tiempo deberá ser empleado en el 20% de las características con mayor prioridad para el cliente
- Los requerimientos deberán incluir pruebas de aceptación las cuales son escritas y verificadas por el cliente (cuando se carece del cliente el líder del proyecto puede ocupar este rol)
- Un alto nivel y diseño visual de la aplicación a desarrollar debe ser producido y debe integrar las mejores prácticas para el diseño de la interfaz de usuario de la plataforma seleccionada

- Codificación por pares
- El código implementara el Modelo-Vista-Controlador (MVC), seguirá un estándar de codificación y será compartido en un repositorio centralizado para que esté disponible a todos los miembros del equipo
- Las pruebas se integran dentro del ciclo de vida en una etapa temprana

Como conclusión, se describieron algunas de las limitantes que existen en los dispositivos móviles, y diversas estrategias para el desarrollo de aplicaciones, por lo cual, las soluciones que ampliamente se utilizan en cómputo tradicional pueden no necesariamente ser una solución y/o tener el mismo impacto. De manera similar, Corral et al. [4] Sugieren que las limitantes de los dispositivos móviles pueden clasificarse en evolutivas, aquellas que serán resueltas en el futuro con la evolución de la misma tecnología y sus recursos; y en limitantes inherentes, restricciones intrínsecas a las plataformas móviles ya que son parte de este sistema, la tarea de aminorar el impacto recae en el desarrollador y las distintas estrategias empleadas. También menciona, que una efectiva estrategia de desarrollo debe ser lo suficientemente fuere para considerar las características y cualidades del ecosistema móvil, incluyendo sus restricciones, se deben tener presentes las expectativas del usuario final y las condiciones del mercado. Al mismo tiempo, debe ser lo suficientemente agresiva y flexible para adaptarse a los avances tecnológicos, sobrellevar la evolución del mercado y la alta competitividad.

## 6. Solución propuesta

Las mejores prácticas que se identificaron al inicio de esta investigación como resultado de un desarrollo empírico de una aplicación en Android, impactan en las fases de: requerimientos, diseño, desarrollo y liberación. Es recomendable considerar también las mejores prácticas propias del lenguaje de programación del ambiente de desarrollo seleccionado, ya que el objetivo de estas, como menciona Withers [16] representan una mejora sobre uno o más aspectos como: productividad, coste, calendario, calidad o satisfacción del usuario.

Algunas de las prácticas de las que se tiene registro y han demostrado su eficacia en otros paradigmas, no necesariamente presentan los mismo resultados para el desarrollo de aplicaciones en Android. Algunas de ellas deben ser redefinidas considerando las restricciones del ambiente móvil. La evolución y aceptación entre la sociedad del ecosistema móvil favorece a la aportación de una propuesta de mejores prácticas para el desarrollo de aplicaciones nativas en Android.

La elección del ambiente de desarrollo a utilizar se debe realizar con base a los requerimientos y el tipo de la aplicación a desarrollar, ya que existen distintas opciones disponibles en la web. Algunos de ellos pueden contar con soporte multiplataforma, o bien, con soporte para un producto en específico, como los videojuegos por ejemplo.

Para el proceso propuesto se ha seleccionado el ambiente de desarrollo PhoneGap, ya que de acuerdo con Babu et al. [36], PhoneGap se ha vuelto popular entre los desarrolladores, por el soporte multiplataforma, el uso de tecnologías web hace que la curva de aprendizaje no sea tan larga en comparación con el manejo del lenguaje del SDK oficial, esto se traduce directamente en un menor costo del proyecto y un menor tiempo para su publicación. Otra característica importante es que permite una mayor portabilidad de la aplicación, su desempeño es comparable al de una aplicación desarrollada a través del SDK oficial y, la interfaz de usuario se beneficia del uso de CSS3 y HTML5 para ser atractiva visualmente.

Esto beneficia directamente a diseñadores, desarrolladores y usuarios por igual, al permitir el uso de patrones de diseño para la creación de la interfaz de usuario, reducir

el tiempo de desarrollo, y la generación de una interfaz de usuario con mayor atractivo visual y funcionalidad.

Finalmente con respecto a las metodologías ágiles. La mayor parte de los principios y buenas prácticas de estas, requieren de cierta experiencia para su empleo. De acuerdo con Scharff [39], es recomendable seguir los siguientes principios de las metodologías ágiles para el diseño de aplicaciones móviles:

- El cliente debe estar constantemente involucrado en el proceso de desarrollo (se puede considerar al usuario final como el cliente ya que es él quien finalmente utilizará la aplicación)
- Los requerimientos deben ser capturados en un alto nivel
- El equipo debe aceptar la realidad de los cambios en los requerimientos
- Los requerimientos deben ser priorizados utilizando la regla de 80/20, es decir, el 80% del tiempo deberá ser empleado en el 20% de las características con mayor prioridad para el cliente
- Los requerimientos deberán incluir pruebas de aceptación las cuales son escritas y verificadas por el cliente (cuando se carece del cliente el líder del proyecto puede ocupar este rol)
- Un alto nivel y diseño visual de la aplicación a desarrollar debe ser producido y debe integrar las mejores prácticas para el diseño de la interfaz de usuario de la plataforma seleccionada
- Codificación por pares
- El código implementara el Modelo-Vista-Controlador (MVC), seguirá un estándar de codificación y será compartido en un repositorio centralizado para que esté disponible a todos los miembros del equipo
- Las pruebas se integran dentro del ciclo de vida en una etapa temprana

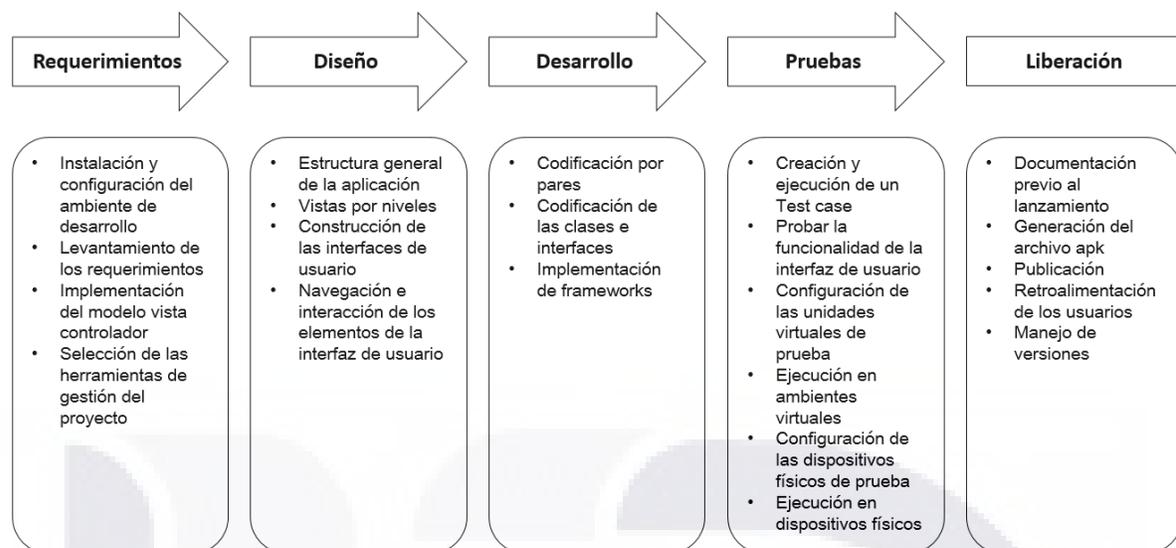
Es posible incluir estos elementos dentro de un modelo de ingeniería de software de cinco etapas (requerimientos, diseño, desarrollo, pruebas y liberación), buscando generar una alternativa integral que permita desarrollar aplicaciones en corto tiempo, centradas en el usuario, haciendo uso de patrones de diseño para añadir simplicidad de mantenimiento y escalabilidad al proyecto. Considerando al cliente (usuario final) en cada etapa del proyecto para realizar pruebas en etapas tempranas de desarrollo, y en

general desarrollar una aplicación que cumpla de inicio a fin con los requerimientos identificados brindando calidad y robustez al producto final.

### **6.1 Modelo de desarrollo de aplicaciones nativas en Android basado en mejores prácticas, metodologías ágiles y elementos del área interacción humano - computadora**

Las diferentes herramientas presentadas anteriormente, buscan reducir el problema de guiar adecuadamente al desarrollador durante cada una de las fases del ciclo de vida del software, para crear una aplicación en poco tiempo, que sea portable, usable para el usuario y con un buen desempeño en general. Es posible integrar las principales características de estas herramientas dentro de un modelo de desarrollo de software, de manera que se pueda dar un correcto seguimiento y evolución de la aplicación y sus requerimientos, identificando las entradas y salidas de cada proceso.

Tomando las bases identificadas de las metodologías ágiles, donde tanto el cliente como la naturaleza cambiante de los requerimientos tienen un papel importante en la calidad del producto final, incluyendo además las mejores prácticas y guías para el diseño de interfaces de usuario, permite que el modelo propuesto sea ajeno e independiente a una plataforma o tecnología en específico, de acuerdo con lo mencionado anteriormente, es una de las principales debilidades en la mayoría de las estrategias actuales.



**Gráfica 23.** Modelo propuesto.

En la Gráfica 23, se presenta la solución propuesta para el problema de desarrollo de aplicaciones nativas en Android independiente de la plataforma, siguiendo un modelo de cinco etapas de desarrollo de software, en las siguientes secciones se describe cada una de estas etapas así como los pasos de dicha solución.

## 6.2 Requerimientos

El éxito de cualquier sistema de software radica en un correcto levantamiento de requerimientos, para poder dar un seguimiento de estos durante cada etapa del ciclo de desarrollo, y de esta manera poder ver cómo el producto va evolucionando, al mismo tiempo es posible determinar el porcentaje de requerimientos validados.

Para el ecosistema de dispositivos móviles, los requerimientos no funcionales toman un mayor valor, ya que están directamente relacionados con el desempeño de la aplicación, manejo de los recursos, consumo de energía, interacción, portabilidad, seguridad, entre otras características presentadas por Wasserman [5].

La selección de los ambientes de desarrollo, el número de integrantes, las estrategias de comunicación que se abordarán con el cliente y con los miembros del equipo, las herramientas de gestión y de desarrollo del producto son una parte crucial de la aplicación. Es una etapa anterior al diseño y construcción de la aplicación, sin embargo, su

selección de acuerdo con la filosofía del equipo de trabajo y de lo que se busca crear, es lo que garantizará el éxito o fracaso de la aplicación.

#### 6.2.1 Instalación y configuración del ambiente de desarrollo

El ambiente de desarrollo que se sugiere en el presente modelo para el desarrollo de aplicaciones nativas en Android, es PhoneGap. Esta elección se debe por las características de portabilidad que brinda en comparación con el ambiente de desarrollo basado en Eclipse, algunas de las cuales se describieron en la sección 5.2.

Sin embargo, el modelo es lo suficientemente flexible como para permitir al equipo cambiar el ambiente de desarrollo por Android Studio, Eclipse SDK, o cualquier otro SDK siempre y cuando estén basados en lenguajes de programación orientados a objetos o en tecnologías web. Por lo tanto, los requerimientos de instalación y configuración para una ambiente en específico deberán consultarse en la web de soporte oficial para dicha plataforma, en la Web de Desarrolladores de Google [31] y PhoneGap [35] respectivamente.

La instalación y configuración de alguna herramienta adicional para PhoneGap [35] puede consultarse directamente en la web oficial. Esta configuración se debe realizar en los equipos de cómputo que se emplearán, de manera que cualquier miembro del equipo cuente con las mismas herramientas y pueda acceder al repositorio de la aplicación para tener la versión más reciente del proyecto. Las herramientas para gestión del proyecto se mencionarán en la sección 6.2.4. Adicionalmente, se recomienda como un paso de configuración crear la estructura de directorios, de manera que al momento de seleccionar la herramienta de configuración del proyecto, solo se sincronicen los archivos con el repositorio.

De acuerdo con Firtman [6], PhoneGap permite tomar un sitio web para dispositivos móviles, es decir, que su configuración ha sido adaptada para mostrar adecuadamente su contenido en un dispositivo con una resolución de pantalla menor en comparación a una computadora; y convertirlo en una aplicación nativa.

De manera similar existen otros frameworks que permiten el desarrollo multiplataforma por medio de una API basada en Javascript, por ejemplo Ionic (<http://ionicframework.com>), y Appcelerator (<http://www.appcelerator.com>). Sin em-

bargo, como menciona Firtman [6], existe hardware y sensores propios de cada compañía de dispositivos móviles al cual no es posible acceder por medio de estas APIs. Algunos ejemplos son Touch ID de Apple [40], que permite identificar a los usuarios por medio de su huella digital, o sensores como el giroscopio, el barómetro, o el sensor de proximidad, presentes en dispositivos de gama alta.

### 6.2.2 Levantamiento de los requerimientos

De acuerdo con Wasserman [5], una correcta lista de los requerimientos no funcionales de la aplicación influye directamente en aspectos de la aplicación como el desempeño, confiabilidad, calidad, y seguridad. Es por ello que los requerimientos funcionales y no funcionales, deben ser capturados y detallados en un alto nivel, para poder dar un correcto seguimiento a lo largo del ciclo de vida del software.

El equipo de desarrollo debe estar consciente de la naturaleza cambiante de los dispositivos móviles, un ejemplo de esto es la implementación de la API de Android en las diferentes versiones que se han liberado hasta el momento, elementos que estaban presentes en las primera versiones han sido cambiados o discontinuados en nuevas versiones, o bien, la mayoría de usuarios prefiere una versión de Android en específico, estos son algunos elementos de portabilidad y escalabilidad que debe tener en cuenta el equipo antes de comenzar con el desarrollo de la aplicación.

El cliente debe estar constantemente involucrado en el proceso de desarrollo, además de que los requerimientos deberán llevar una prueba de aceptación por escrito y serán verificados por el cliente, de acuerdo con los principios de las metodologías ágiles presentados por Scharff [39]. Si no se cuenta explícitamente con un cliente, una estrategia es considerar al usuario final en su lugar. Dicha estrategia se utiliza con aplicaciones de carácter general donde el equipo de desarrollo busca crear una aplicación, entonces, considerar qué aspectos o características buscan los usuarios como los requerimientos del cliente es una opción viable.

### 6.2.3 Implementación del modelo vista – controlador

En la ingeniería de software, existen diferentes herramientas que permiten representar los conceptos e ideas más complejas de tal manera, que se facilite la comunicación entre los miembros del proyecto. El uso de modelos permite entender y representar el problema a resolver así como su solución. La implementación del modelo vista – con-

TESIS TESIS TESIS TESIS TESIS

trolador (MVC) ampliamente utilizado en los grandes sistemas de software para cómputo tradicional, puede adaptarse al contexto de los dispositivos móviles, ya que permite dividir el sistema en tres componentes: el modelo, donde se encuentra la lógica del sistema, la vista, que representa la interfaz de usuario, y por último, el control, que gestiona las solicitudes entre ambas capas, de acuerdo con Scharff [39].

De acuerdo con Selfa et al. [41], el uso del MVC ha demostrado sus beneficios para aplicaciones interactivas, ya que permite múltiples representaciones de la misma información, promueve la reutilización de código, ayuda a que los desarrolladores se concentren en un solo aspecto de la aplicación. Estas características facilitan el mantenimiento, agregan escalabilidad a la aplicación y en general, una mayor cohesión entre sus componentes.

La configuración de los ambientes de desarrollo permite la implementación del MVC, además, de que dicha arquitectura no está dirigida a una tecnología en específico, por lo tanto es posible el desarrollo de una aplicación nativa en Android a través del uso de tecnologías web.

#### 6.2.4 Selección de las herramientas de gestión del proyecto

Las herramientas para la gestión del proyecto permiten al equipo llevar un mejor control de los roles de cada integrante, las acciones que se realizan de manera individual y/o grupal. Scharff [39] menciona la importancia de un repositorio centralizado, donde se resguarda el proyecto y su documentación, de manera que cada miembro pueda descargar una copia del código fuente, trabajar en ella de manera local, y posteriormente sincronizar con el repositorio central.

GitHub [42] es un repositorio de código abierto, permite dar un seguimiento a los errores existentes al asignarse a uno o más miembros del equipo, consultar su estado, etiquetarlos y comentarlos. Los miembros del equipo pueden tener acceso de solo lectura, lectura y escritura, o acceso como administradores a los repositorios de acuerdo con su rol dentro del proyecto. Por estas características es considerada una red social para el manejo de repositorios.

Dabbish et al. [43] presentan algunas de las ventajas del uso de herramientas de cómputo social en el desarrollo de software, ya que constituyen una ventaja para la

administración del proyecto, es posible identificar a cada persona dentro del proyecto por medio de sus perfiles, se realiza un seguimiento de los artefactos generados durante el ciclo de vida, el equipo de trabajo es consciente de las acciones implementadas de manera individual y/o grupal, mantiene una copia de seguridad del código fuente y permite al equipo trabajar en diferentes ubicaciones.

Tanto PhoneGap como Android Studio, promueven el uso de esta herramienta para la gestión del proyecto, de manera que se facilita consultar ejemplos de códigos disponibles en ambas plataformas, y en el caso de PhoneGap, solo se indica la dirección del repositorio del proyecto previo a generar el código para ejecutar la aplicación nativa en Android.

### **6.3 Diseño**

Un alto nivel y diseño visual de la aplicación a desarrollar deben producirse en base a los requerimientos capturados, además, el diseño de la interfaz debe integrar las mejores prácticas para el diseño de interfaces de usuario para la plataforma seleccionada.

El diseño de la interfaz de usuario debe considerar también la portabilidad de la aplicación a desarrollar, es decir, la portabilidad es la habilidad de aplicación para ejecutarse en distintas plataformas con la misma funcionalidad. Hay que tener en cuenta que el sistema operativo Android se puede ejecutar en distintos dispositivos como relojes inteligentes, celulares de nueva generación, tabletas, etc., y a su vez, estos vienen en una extensa gama de resoluciones de pantalla.

Los patrones de diseño como lo menciona Thung et al. [3] representan una solución para problemas similares durante la fase de diseño, promueven la reutilización de código y mejoran la usabilidad de la aplicación. La selección de los patrones de diseño debe realizarse en base a los requerimientos, de esta manera, se podrá determinar qué patrones utilizar y cómo integrarlos dentro de la aplicación, un ejemplo de esto es la sección de patrones de diseño [44] que se presentan en la Web de Desarrolladores de Google.

### 6.3.1 Estructura general de la aplicación

Las aplicaciones móviles pueden tener tantas pantallas como se requiera, en base a su contenido y el detalle o desglose de cada una de estas vistas. Aplicaciones como una calculadora o la cámara, por lo general, sólo utilizan una pantalla principal, en cambio, aplicaciones como asistentes de correo, con un contenido más amplio y una navegación más profunda, requieren de un mayor número de pantallas y vistas.

La estructura general de la aplicación se puede representar por medio de un árbol jerárquico, de manera que se puedan identificar las vistas principales y las vistas detalladas. Para esto, es recomendable considerar el contenido que se mostrará en cada pantalla, utilizar los diferentes elementos de la API de Android para tener una navegación consistente, e identificar como el usuario interactuara con cada uno de ellos. El árbol jerárquico debe considerar la navegación con el dispositivo en posición horizontal y en vertical, y considerar las mejores prácticas o patrones de diseño para permitir una navegación fluida entre las vistas principales y las vistas detalladas, como se menciona en la sección de patrones de diseño dentro de la Web de Desarrolladores de Google [44].

### 6.3.2 Vista por niveles

De acuerdo con lo presentado por Google [44], el diseño de la interfaz de usuario se puede dividir en 2 niveles que son, vistas de alto nivel y vistas detalladas. Las vistas de alto nivel, son las vistas principales que soporta la aplicación. Estas vistas representan la misma información y en conjunto, muestran la funcionalidad y navegabilidad de la aplicación. Por otra parte, las vistas detalladas muestran esta información utilizando los diferentes elementos de la API, de manera que sean visualmente atractivas para el usuario.

Las vistas de alto nivel son también la primera pantalla que se presenta al usuario, como una pantalla de inicio de sesión o un menú de categorías con diferentes opciones a elegir. El uso de árboles de jerarquías para representar las vistas de la aplicación en una opción para el diseño, permite identificar las vistas con mayor información, de manera que la selección de los archivos de diseño sea de acuerdo con la información a presentar, los elementos de navegación a utilizar y los requisitos de navegabilidad de la aplicación.

Las vistas detalladas son aquellas donde se presenta información específica al usuario, de manera que le permita consultar, editar o crear cambios en ella. Estas vistas no son necesariamente independientes, ya que, si el usuario requiere observar distintos elementos en secuencia, las vistas detalladas deben permitir la navegación entre estos elementos, de manera que el usuario no debe regresar a un nivel superior, sino que se tenga una navegación fluida entre las vistas detalladas por medio de los elementos de la interfaz de usuario seleccionados para su construcción.

### 6.3.3 Construcción de las interfaces de usuario

El diseño de las interfaces de usuario requiere un estudio de los elementos disponibles en las diferentes versiones de la API de Android, esto se debe a que existen elementos particulares de versiones anteriores del sistema operativo que ya no fueron tomados en cuenta en futuras versiones. La funcionalidad y usabilidad de la aplicación debe ser la misma, sin importar el dispositivo móvil o la versión del sistema donde se ejecute la aplicación.

Los patrones de diseño permiten la reusabilidad y repetición de las soluciones en problemas similares durante el proceso de diseño. De acuerdo con lo mencionado por Thung et al. [3], los patrones de diseño son una solución general a un problema de particular de diseño, lo que crea una base común para resolver problemas similares, fomentando la reusabilidad y mejorando la productividad del equipo de diseño. Además, el uso de patrones de diseño ha probado mejorar la usabilidad de la aplicación, su éxito y la reducción de tiempo entre los usuarios para familiarizarse con el sistema.

La Web de Desarrolladores de Google cuenta con una sección que describe los patrones de diseño disponibles en Android [44]. Algunos de los patrones que se describen son: selección, confirmación, transición de las vistas, pantalla completa, por mencionar algunos ejemplos. Es recomendable el análisis de cada uno de ellos, de manera que se seleccionen solo aquellos patrones que se adecuen con los requerimientos de la aplicación. Esto se refleja en la navegabilidad entre las diferentes vistas de la aplicación.

### 6.3.4 Navegación e interacción de los elemento de la interfaz de usuario

El último paso dentro de la fase de diseño es la evaluación de las interfaces de usuario y su navegación. De acuerdo con Scharff [39], las pruebas se deben integrar dentro del ciclo de vida en una etapa temprana. Al final de la fase de diseño no se ha realizado

aun algún tipo de codificación, sin embargo, las vistas del sistema han sido terminadas y deben probarse con el cliente si cubren los requisitos de navegación que fueron acordados al inicio.

Las pruebas de la interfaz se realizan en usuarios que son ajenos al proyecto, otro enfoque como el presentado por Kaasila et al. [45], sugiere el uso de scripts en herramientas automatizadas buscando un análisis más exhaustivo. Las ventajas del uso de herramientas automatizadas son que permiten la prueba en paralelo en diferentes dispositivos Android, a su vez, estos con diferentes versiones del sistema operativo, lo que permite identificar aquellas plataformas donde se presentan un mayor número de errores o fallos.

## 6.4 Desarrollo

La fase de codificación de la aplicación debe considerar el uso de estrategias que permitan una administración adecuada de los recursos limitados del ecosistema móvil. La implementación de estrategias como el uso de mejores prácticas, se ve reflejado en la calidad, desempeño, uso de la batería y memoria del dispositivo móvil, y funcionalidad como presentan Withers [16] y la Web de Desarrolladores de Google [31] respectivamente.

Sin embargo, se debe evaluar cuales prácticas se utilizarán en función de los requerimientos, la plataforma para desarrollo y las herramientas utilizadas para crear la aplicación móvil.

### 6.4.1 Codificación por pares

De acuerdo con Williams [46], la programación en pares es un estilo de programación en el cual 2 programadores trabajan lado a lado en una misma computadora, continuamente colaborando en el diseño, algoritmo, código y/o pruebas. Uno de los pares, llamado el "conductor" es quien escribe en la computadora o realiza el diseño. Su socio, llamado el "navegador", tiene diferentes trabajos. Uno es observar el trabajo del conductor, buscando defectos. Juntos, el conductor y el navegador continuamente diseñan una solución, y de manera periódica intercambian sus roles.

La programación por pares debe considerar un ambiente controlado, donde los participantes están familiarizados con esta estrategia y comprometidos con el proyecto, se realizan evaluaciones individuales y por equipo de su rendimiento, además, el líder de proyecto está al tanto de las fortalezas y debilidades del equipo, para asistirlos cuando sea necesario.

#### 6.4.2 Codificación de las clases e interfaces

Scharff [39] sugiere además, el uso de un estándar de codificación el cual debe implementarse de manera prudente, y ser práctico. El código fuente del proyecto tendrá así un estilo armonioso, sin importar el número de desarrolladores involucrados; cualquier otro desarrollador dentro del proyecto será capaz de manipularlo.

El uso de mejores prácticas para la codificación resulta una opción ideal para el manejo de recursos y un mejor desempeño de la aplicación. De acuerdo con Withers [16], una mejor práctica puede ser una práctica técnica o de gestión, que ha demostrado una mejora sobre uno o más aspectos como: productividad, coste, calendario, calidad o satisfacción del usuario.

Los desarrolladores pueden encontrar información sobre mejores prácticas para la plataforma que estén utilizando en las web de desarrollo oficial, en los estándares de codificación y en la literatura académica como lo presentado por Perks [17], Tonini et al. [1], y Oh et al. [2]. Sin embargo, se debe discernir cuáles se pueden incorporar al proyecto de aquellas que no aportarían en absoluto.

De acuerdo con Wasserman [5], las aplicaciones móviles pueden llegar a ser productos demasiado complejos, por lo que es conveniente dividir el problema general en subproblemas, y a su vez, estos deben ser afrontados con la información disponible que mejor se adecúe con los requerimientos no funcionales del sistema.

#### 6.4.3 Implementación de los frameworks

De acuerdo con Johnson & Foote [47], un framework se define como un conjunto de clases que incorporan un diseño abstracto para solucionar una familia de problemas relacionados. Partiendo de esta definición, y como se mencionó anteriormente en la sección 5.2 por Firtman [6], PhogeGap constituye un framework para el desarrollo de

TESIS TESIS TESIS TESIS TESIS

aplicaciones nativas en Android, cuenta con una API y métodos propios para su implementación.

De manera similar, Android Studio como otros ambientes de desarrollo disponible, cuentan con las características para incorporar el uso de frameworks, de manera que se busca aumentar la funcionalidad y mantenimiento del código de la aplicación. Es recomendable referirse a la web oficial de cada uno de estos frameworks para consultar información relacionada con su instalación, uso y mantenimiento, como es el caso de la Web de Desarrolladores de Google [31] para Android Studio, y PhoneGap [35] para el framework respectivamente.

### 6.5 Pruebas

Escribir y ejecutar pruebas es parte esencial dentro del ciclo de desarrollo de una aplicación de acuerdo con lo presentado en la Web de Desarrolladores de Google [31]. De la misma manera, Scharff [39], menciona la importancia de integrar las pruebas dentro del ciclo de vida en una etapa temprana. Lo que permite identificar errores y corregirlos para mejorar la calidad del producto final.

De la misma manera en que existen diversas estrategias para el desarrollo de aplicaciones en Android, también existen diferentes alternativas para probar y evaluar el producto. El sistema operativo Android consta de diversos componentes como presenta Jackson [22], los cuales son Actividades, Servicios, Receptores de transmisiones, y Proveedores de contenidos. Por lo que cada componente debe administrarse en base a un conjunto de reglas y normas adecuadas. De acuerdo con Hu & Neamtiu [48], los errores más comunes que se presentan en las aplicaciones son debidos a un incorrecto manejo de las excepciones en el código, no se valida correctamente las entradas y/o salidas de los campos de texto, un manejo incorrecto de los componentes de la API, por mencionar algunos ejemplos. Por lo tanto, la interfaz y el código deben evaluarse individualmente y en conjunto para prevenir cualquier error, con el objetivo de entregar un producto con un adecuado nivel de calidad.

### 6.5.1 Creación y ejecución de un caso de prueba

Un caso de prueba es una serie de pasos que se realizan en un escenario controlado, de manera que se puede identificar la salida o resultado esperado en base a las entradas realizadas.

Un caso de prueba define un conjunto de objetos y métodos para ejecutar múltiples pruebas independientes una de otra. Los casos de prueba pueden ejecutarse de manera programada, es decir, a una hora en específico y repitiendo una serie de pasos, o bien, pueden ejecutarse sobre demanda, en función del framework y herramientas que se utilicen, de acuerdo a información proporcionada en el apartado de pruebas de la Web de Desarrolladores de Google [49].

Los casos de prueba se utilizan para verificar que no hay errores en el diseño, o en el comportamiento funcional de la aplicación. La creación de los casos de prueba puede variar de acuerdo a la plataforma de desarrollo que se ha seleccionado, se recomienda consultar la web oficial de cada plataforma, la Web de Desarrolladores de Google [31], y la web de PhoneGap [35] respectivamente. O bien, en caso de utilizar una herramienta automatizada, de acuerdo con Kaasila et al. [45], y Hu & Neamtiu [48], se deben considerar los requerimientos necesarios para su utilización.

### 6.5.2 Probar la funcionalidad de la interfaz de usuario

Anteriormente en la sección 6.3.4 se ha probado la interfaz de usuario, sin embargo, las pruebas que se realizaron en ese punto no incluían aun código o procedimientos. Después de la fase de desarrollo donde se han codificado las clases, interfaces y se han implementado los frameworks, se debe evaluar nuevamente la interfaz de usuario para verificar que los cambios realizados en la fase de desarrollo no han modificado su funcionamiento.

La interfaz gráfica del usuario (GUI por sus siglas en inglés) de una aplicación Android consta de una estructura jerárquica como se presentó en la sección 6.3.2, donde las vistas aceptan entradas generadas por el usuario, a su vez, estas generan diferentes eventos dentro del sistema y producen una salida gráfica. Cada GUI se conforma de distintos objetos, los cuales tienen propiedades y atributos definidos en la fase de diseño, y los cuales pueden cambiar durante la ejecución, como presenta Memon [50].

En la literatura se han reportado diferentes enfoques sobre el uso de herramientas automatizadas para evaluación de la GUI. Kaasila et al. [45] sugieren el uso de scripts en herramientas automatizadas buscando un análisis más exhaustivo de la GUI. Amalfitano et al. [51] proponen el uso de herramientas como AndroidRipper para analizar la GUI de la aplicación, haciendo uso de los casos de prueba para su operación.

### 6.5.3 Configuración de las unidades virtuales de prueba

Un dispositivo virtual Android (AVD, por sus siglas en inglés) es un emulador que permite modelar un dispositivo al definir sus características de hardware, y software para ser emulados por el emulador de Android. Esta funcionalidad se encuentra presente en el SDK oficial, además de indicar los pasos para su configuración en el apartado de pruebas de la Web de Desarrolladores de Google [49]. El uso de ambientes virtuales de prueba representa una opción viable cuando no se puede costear la adquisición de diferentes dispositivos físicos para probar directamente el desempeño, y funcionalidad de las aplicaciones.

Un AVD se conforma de: (1) un perfil de hardware, define las características de hardware que tendrá el dispositivo, como los sensores, tipo de teclado, capacidad de memoria, entre otros. (2) Una imagen del sistema, la versión del sistema operativo Android que se ejecutara en el dispositivo virtual. (3) Otras características, como la apariencia, tamaño de la pantalla, emular el uso de una tarjeta de memoria SD. Y finalmente, (4) un área dedicada al almacenamiento, para almacenar información propia del sistema y de las aplicaciones instaladas.

En la literatura se han reportado el uso de herramientas de código abierto para emular el comportamiento, y funcionalidad de sensores que no están presentes en el emulador oficial de Android. Serfass et al. [52] muestran el uso de una de estas herramientas para emular los sensores inalámbricos y la comunicación puerto a puerto por medio de NFC (Near Field Communication).

### 6.5.4 Ejecución en ambientes virtuales

Se debe mencionar que ejecutar un AVD requerirá de memoria, y espacio de almacenamiento del computador donde se va a ejecutar el emulador. Por lo que dependiendo de las características de hardware y software del computador, se puede esperar un rendimiento menor en comparación con un dispositivo físico.

Una alternativa es el uso de dispositivos virtuales sobre IP, como describen Chen & Itoh [53]. Donde los usuarios controlan imágenes de sus dispositivos virtuales, por medio de un cliente instalado en sus dispositivos (e.g. smartphones). Esto permite a las aplicaciones instalarse y ejecutarse en la nube, emulando el comportamiento en el dispositivo local, integrando características de un ambiente remoto dentro de un ambiente local.

Los casos de prueba creados deben ejecutarse de principio a fin, realizando las mismas acciones en cada ocasión, independientemente si la ejecución es en un ambiente virtual o físico. Los errores encontrados, el nivel de desempeño y demás información recabada debe preferentemente compararse con los resultados obtenidos en un dispositivo físico para su validación.

#### 6.5.5 Configuración de los dispositivos físicos de prueba

Cualquier dispositivo con el sistema operativo Android puede utilizarse para ejecutar, depurar y probar aplicaciones. Las herramientas y servicios que utilizan los ambientes de desarrollo como Android Studio, PhoneGap y otros ambientes de código abierto, cuentan con características para manipular el código fuente y permitir su instalación y ejecución en dispositivos físicos como presentan Firtman [6], la Web de Desarrolladores de Google [31], y el sitio oficial de PhoneGap [35] respectivamente.

Es recomendable probar la aplicación en la mayor cantidad de dispositivos físicos disponibles, por ello una alternativa a la adquisición de dichos equipos es el uso de terminales físicas de manera remota. En la literatura se ha reportado el uso de herramientas para probar el funcionamiento de las interfaces de usuario, como lo presentaron Kaasila et al. [45] con testdroid. De manera similar, diferentes compañías ofrecen este servicio sin cargo, por un periodo de prueba, o bien, con un cargo, dependiendo del tiempo de sesión y número de terminales a utilizar, como es el caso de Vodafone [54], Testdroid [55], y Keynote [56] respectivamente.

Al igual que con el uso de dispositivos virtuales, lo recomendable es probar la aplicación, ejecutar los casos de prueba en la mayor cantidad de dispositivos posibles, buscando identificar los errores, corregirlos y mejorar el desempeño de la aplicación en beneficio de los usuarios finales.

#### 6.5.6 Ejecución en dispositivos físicos

Los resultados de los casos de prueba en ambientes virtuales se deben comparar con los obtenidos en ambientes físicos, de manera que los errores reportados sean repetibles e identificables en ambos escenarios.

Para verificar que la aplicación cumple con los requerimientos funcionales y no funcionales, se debe asegurar que el código no tenga problemas de estructuración. Un código no estructurado puede impactar en la confiabilidad y eficiencia de la aplicación, haciendo que se dificulte su mantenimiento, como se ha reportado en la Web de Desarrolladores de Google en el apartado de pruebas [49].

Por ello, Wasserman [5] resalta el hecho de utilizar las mejores prácticas que mejor se adecuen a la complejidad del problema a resolver. Adicionalmente Scharff [39], menciona la importancia de utilizar un repositorio centralizado y actualizado, junto con un estándar de codificación que permitan dar mantenimiento y escalabilidad al código.

### 6.6 Liberación

La liberación de un producto es el paso final para presentar al público el trabajo que se ha desarrollado, haciendo uso de medios oficiales como Google Play, u otras plataformas en función del sistema operativo para el que se esté desarrollando.

En esta etapa, la aplicación esta lista para publicarse, por lo que se debe integrar la documentación que pueda requerir el usuario final, así como definir las estrategias para su publicación y distribución entre el público en general.

#### 6.6.1 Documentación previa al lanzamiento

Adicional a la información que pueda requerir el usuario para guiarlo en el funcionamiento de la aplicación desarrollada, existen varios aspectos que deben entenderse y definirse de ser necesario antes de publicar la aplicación en Google Play. A continuación se mencionan los principales aspectos que debe tener presente el equipo de desarrollo previo a la publicación del producto.

El equipo de desarrollo debe asegurar de entender y seguir las políticas del programa de Google Play [57], las cuales se deben aceptar al momento de registrarse. Google Play hace cumplir activamente sus políticas y cualquier violación a estas, puede derivar

TESIS TESIS TESIS TESIS TESIS

en la suspensión de las aplicaciones, o incluso, en la cancelación de la cuenta como desarrollador.

Las pruebas realizadas en la sección 6.5 permiten comprobar que los requerimientos de la aplicación satisfacen las necesidades del cliente. Adicional a estas pruebas, es recomendable verificar que se cumplen los criterios de calidad para las aplicaciones mencionados en la sección Core App Quality, dentro de la Web de Desarrolladores de Google [31].

Se define el contenido de la aplicación, es decir, el nivel de madurez del contenido que ofrece la aplicación. Los niveles disponibles que ofrece Google Play son: para todos, nivel de madurez bajo, nivel de madurez medio, y nivel de madurez alto.

La importancia de identificar en que países se podrá descargar la publicación se debe principalmente a: los recursos externos que solicita la aplicación y que estos puedan estar o no disponibles en el país seleccionado, proporcionar el contenido en el idioma oficial de los países seleccionados, los requerimientos legales que implique el uso y descarga de la aplicación de acuerdo a la legislación de cada país, la diferencia de precio, ente otros.

Se agrega información sobre en qué plataformas, o versiones del sistema operativo Android se puede ejecutar la aplicación. Esta información se define en el archivo AndroidManifest dentro de la aplicación, de acuerdo por lo presentado en la Web de Desarrolladores de Google [31]. Adicionalmente, se confirma cuales dispositivos y resoluciones de pantalla soportan la aplicación, incluyendo algunas capturas de pantalla.

Adicional al diseño de la aplicación, existen aspectos a considerar previos a la publicación como son: el nombre de la aplicación, el icono de descarga, los textos promocionales, las capturas de pantalla, y los videos que pudieran utilizarse para mostrar el producto a los usuarios. Ya que esta información estará presente en la ficha de descarga de Google Play, y en base a su contenido el usuario decide si descargará o no la aplicación.

#### 6.6.2 Generación del archivo apk

El tamaño total de la aplicación puede afectar su diseño y distribución. Actualmente, el tamaño máximo permitido en Google Play para un apk es de 50 MB, como se menciona

en la Web de Desarrolladores de Google [31]. Similarmente, Google Play ofrece servicios para almacenar apks como archivos de expansión, de manera, que sea posible utilizar archivos de hasta 2 GB, o bien, por otro lado, existen herramientas disponibles que analizan el código de la aplicación de manera que reducen el tamaño final del apk generado.

Para mayor información sobre la generación del archivo apk de acuerdo con la plataforma de desarrollo utilizada, es recomendable dirigirse a la web de desarrollo oficial donde se encuentra información al respecto, como es el caso de la Web de Desarrolladores de Google [31] para Android Studio, y PhoneGap [35] para el framework respectivamente.

### 6.6.3 Publicación

Uno de los beneficios del uso de la plataforma oficial Google Play para la distribución de aplicaciones Android, es el hecho de que los productos pueden estar disponibles en todos los países donde Google Play tiene presencia. Como se mencionó anteriormente en la sección 6.6.1, definir en qué países se podrá descargar la aplicación para estandarizar el precio de comercialización. En la sección Distribución dentro de la Web de Desarrolladores de Google [31] se presentan los servicios, requerimientos, y estrategias disponibles para comercializar aplicaciones. Por otra parte, como se mencionan en la web oficial, decidir si las aplicaciones serán gratuitas o de pago es importante porque, en Google Play las aplicaciones gratuitas deben permanecer gratuitas.

Una vez que se tiene listo el archivo apk el equipo de desarrollo puede elegir entre publicar directamente o bien probar la aplicación antes de publicarla a través de la consola para desarrolladores. Google Play permite definir un grupo de usuarios que probaran la aplicación, conocidos como alpha o beta testers, los cuales se pueden encontrar en cualquier parte del mundo. Es importante mencionar que los usuarios en las versiones alpha o beta no podrán dejar comentarios de ningún tipo o evaluaciones de la aplicación en la ficha de descarga.

### 6.6.4 Retroalimentación de los usuarios

Después de publicar o actualizar la aplicación es recomendable atender a los comentarios y valoraciones de los usuarios. La consola para desarrolladores de Google [31] permite llevar un control por cada aplicación publicada sobre los errores reportados,

valoraciones y opiniones de los usuarios, estadísticas de las instalaciones y desinstalaciones diarias y totales, entre otras características.

Se debe definir una estrategia para dar un correcto seguimiento a la aplicación una vez que se ha publicado, atender a los comentarios de los usuarios y de ser necesario, tomar nota de los errores reportados para corregirlos en una próxima actualización.

#### 6.6.5 Manejo de versiones

Al momento de liberar una actualización de la aplicación no es necesario seguir los pasos para liberar un nuevo producto. De acuerdo con Dabbish et al. [43], se crea una ramificación del proyecto principal para realizar los cambios necesarios en el código corrigiendo los errores reportados, una vez que han sido validados por el cliente o líder del proyecto de manera local, se podrá entonces sincronizar con el repositorio central.

Al generarse nuevamente un archivo apk se debe cerciorar de que se cumplen los siguientes requisitos: el nombre del paquete del apk actualizado debe ser el mismo que el de la versión actual, la versión del código debe ser mayor que la versión actual, se puede consultar en el archivo AndroidManifest, finalmente, el apk actualizado debe firmarse con la misma firma digital con que se firmó la versión actual publicada en Google Play.

Al momento en que se reemplaza el apk por el nuevo dentro de la consola de desarrolladores, se notifica en automático a los usuarios de que existe una versión actualizada de la aplicación, se recomienda atender la misma estrategia utilizada para el seguimiento de la retroalimentación de los usuarios para mejorar la calidad del producto y en beneficio de los usuarios finales.

## 7. Caso de estudio

El caso de estudio consiste en presentar el modelo para el desarrollo de aplicaciones nativas en Android, junto con la descripción de cada una de sus etapas, a un grupo de estudiantes con los conocimientos suficientes en las áreas de: 1) desarrollo de software a través de lenguajes de programación orientados a objetos, 2) diseño de páginas web adaptadas para dispositivos móviles, 3) Android, y 4) procesos de Ingeniería de Software.

La información que se presentará a los alumnos comprende en su totalidad el capítulo anterior de esta tesis, donde se reportan los hallazgos de las distintas áreas consideradas y se incorporan dentro de un modelo de desarrollo de software, lo cual constituye la propuesta de la investigación.

Los objetivos del caso de estudio son los siguientes:

- Medir empíricamente el concepto del modelo de desarrollo de aplicaciones nativas en Android basado en mejores prácticas y metodologías ágiles.
- Medir la percepción de los estudiantes sobre el modelo diseñado en cuanto a usabilidad y adopción.

Para lograr estos objetivos es necesaria la participación de estudiantes de la Universidad Autónoma de Aguascalientes, de la Licenciatura en Ingeniería en Computación Inteligente. Por ello, se revisó el plan de estudios de esta carrera para identificar a partir de qué semestre los alumnos cuentan con los conocimientos suficientes en las áreas anteriormente mencionadas.

Los estudiantes considerados para participar voluntariamente en el estudio son de 5to semestre en adelante. Esta decisión se toma con base a una revisión del plan de estudios, ya que se considera que los alumnos han desarrollado las habilidades necesarias para utilizar las diferentes metodologías de la ingeniería de software para realizar proyectos de desarrollo de software. Además, cuentan con los conocimientos para analizar y evaluar las técnicas y metodologías de la ingeniería de software con el propósito de desarrollar software de calidad y que pueda competir a nivel tecnológico, así como también, sintetizar los métodos y técnicas en la interacción humano – computadora

para proporcionar productos de software más amigables. Esto de acuerdo a lo reportado en el plan de estudios de la licenciatura.

Con la finalidad de evaluar el concepto, la usabilidad y adopción del modelo propuesto en un solo instrumento de evaluación, se diseñó el siguiente cuestionario con 12 sentencias. Dicho cuestionario se presentará a los participantes del estudio al término del análisis de la información descrita del modelo para el desarrollo de aplicaciones nativas en Android.

**Tabla 5.** Cuestionario para evaluación del proceso.

**1. El proceso para el diseño de aplicaciones nativas en Android basado en mejores prácticas y metodologías ágiles, cuenta con un adecuado soporte científico/teórico.**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**2. El marco teórico utilizado es relevante para el diseño de aplicaciones nativas en Android**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**3. Cada una de la etapas del proceso son coherentes**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**4. El proceso es adecuado para lograr el propósito de su utilización**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**5. Si utilizara el proceso, podría lograr diseñar una aplicación nativa en Android de manera más eficiente**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**6. El proceso es complejo, y difícil de utilizar**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**7. Si utilizara el proceso, simplificaría el trabajo de diseñar aplicaciones nativas en Android**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**8. Considera que aprender a usar el proceso, es sencillo**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**9. Si usara el proceso, le resultaría fácil de adaptar a sus necesidades**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**10. Considera que proceso sería difícil de usar en la práctica**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de
--------------------------	---------------	-------------	------------	---------------

desacuerdo				acuerdo
<b>11. Consideraría utilizar el proceso presentado en futuros proyectos para desarrollar aplicaciones nativas en Android</b>				
Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
<b>12. Preferiría utilizar otras estrategias para diseñar aplicaciones Android, antes que utilizar el proceso presentado</b>				
Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo

El cuestionario permite evaluar tres criterios, los cuales son: concepto, usabilidad y aceptación. Las preguntas de la 1 a la 4 miden conceptualmente el proceso, de la 5 a la 10 miden la usabilidad, finalmente las últimas dos preguntas servirán para medir la adopción.

Cada una de las diferentes opciones de respuesta tienen un valor asignado, el cual es: 0 para "Fuertemente en desacuerdo", 1 para "En desacuerdo", 2 "Indiferente", 3 "De acuerdo", y 4 puntos para "Totalmente de acuerdo". Cabe mencionar que las preguntas 6, 10 y 12, tienen una valoración invertida, es decir, 4 para "Fuertemente en desacuerdo", 3 para "En desacuerdo", 2 "Indiferente", 1 "De acuerdo", y 0 puntos para "Totalmente de acuerdo".

Al final del caso de estudio, de acuerdo al número de participantes se contabilizarán sus respuestas para cada uno de las sentencias del cuestionario. El criterio de evaluación se presenta en la Tabla 6, donde para cada una de los objetivos planteados se debe obtener una calificación general de al menos el 80% de los puntos totales.

Para que los resultados puedan ser considerados como un éxito, el 80% de los participantes deben obtener al menos el 80% de los puntos en cada una de las categorías mencionadas. En la Tabla 6, se presentan los criterios utilizados para medir cada uno de los objetivos planteados.

**Tabla 6.** Métricas de medición.

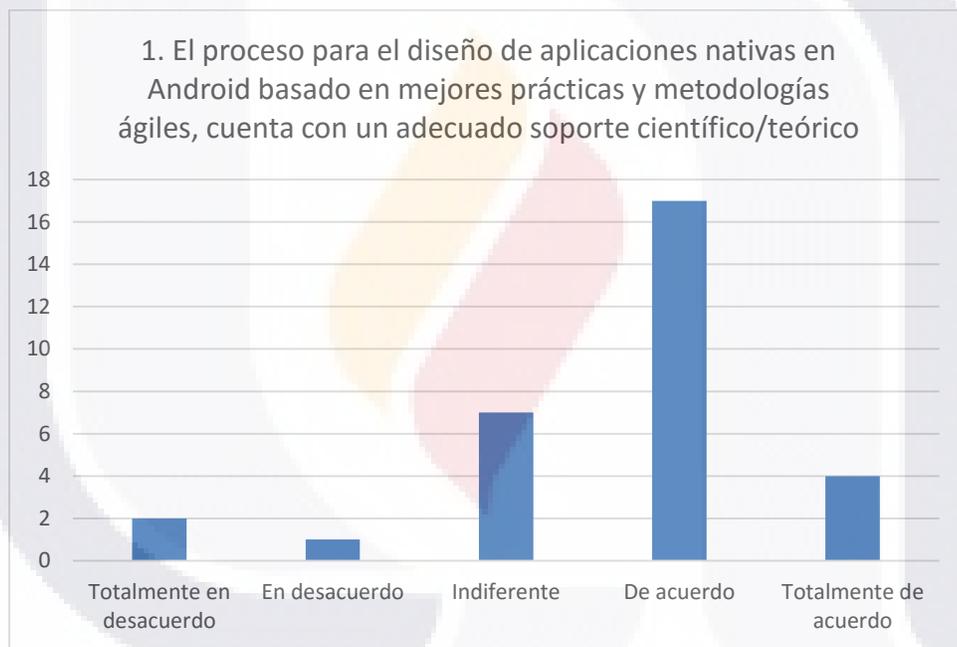
Objetivo de medición	Porcentaje	Equivalencia
<b>Concepto</b>	80%	12.8 puntos
<b>Usabilidad</b>	80%	19.2 puntos
<b>Aceptación</b>	80%	6.4 puntos

Esta clasificación permite evaluar por separado cada uno de los objetivos de medición, y con esto identificar oportunamente las áreas de oportunidad del modelo propuesto. Adicionalmente, permite determinar si la hipótesis de la tesis es válida, al obtener una aprobación por al menos el 80% del total de participantes en el estudio. En caso contrario de no obtener tal porcentaje, se determinará el por qué el proceso no representa una solución para el desarrollo de aplicaciones nativas en Android.



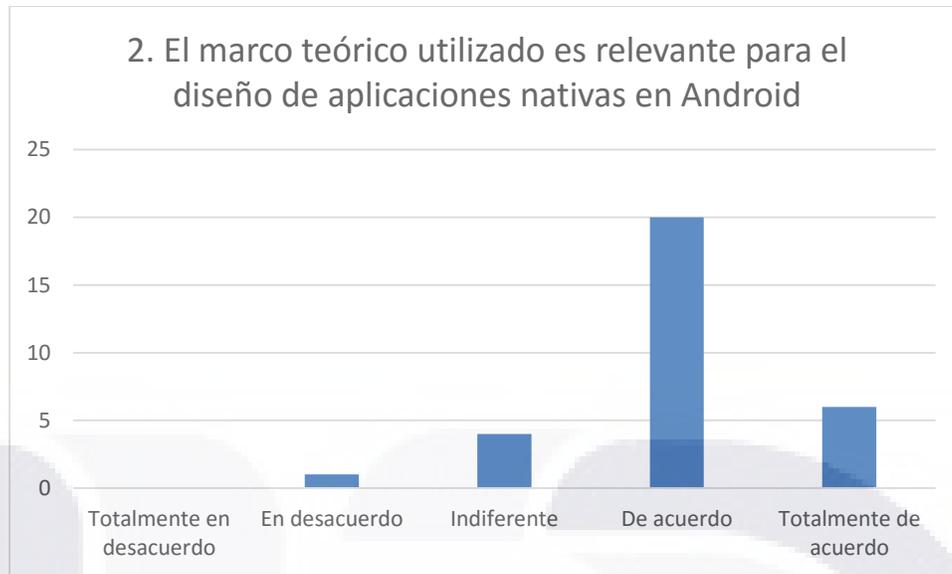
## 8. Resultados

El caso de estudio contó con la participación de 31 estudiantes de la licenciatura en Ingeniería en Computación Inteligente, los resultados que se presentan a continuación se desglosan por pregunta, mostrando así la percepción general de los estudiantes sobre el proceso propuesto. Se consideran como resultados positivos la selección de cualquiera de las opciones “De acuerdo” y “Totalmente de acuerdo”. Seleccionar “Indiferente” es de carácter neutral, y finalmente la selección de las opciones “Totalmente en desacuerdo” y “En desacuerdo” son resultados negativos. Cabe mencionar que las sentencias 6, 10 y 12 tienen una ponderación a la inversa, es decir, seleccionar opciones *negativas*, es en realidad positivo para los fines del caso de estudio.



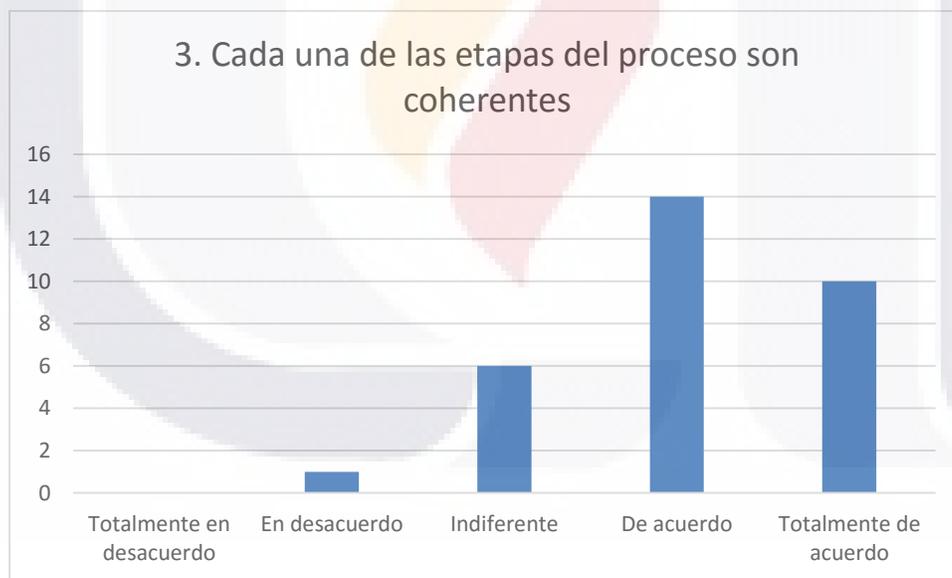
**Gráfica 24.** Primera pregunta del instrumento de evaluación.

21 de los participantes mostraron su aprobación con respecto a que el modelo propuesto cuenta con un adecuado soporte científico y teórico. Tan solo 7 se mostraron indiferentes, y solo 3 consideraron que es necesario mejorar la literatura consultada.



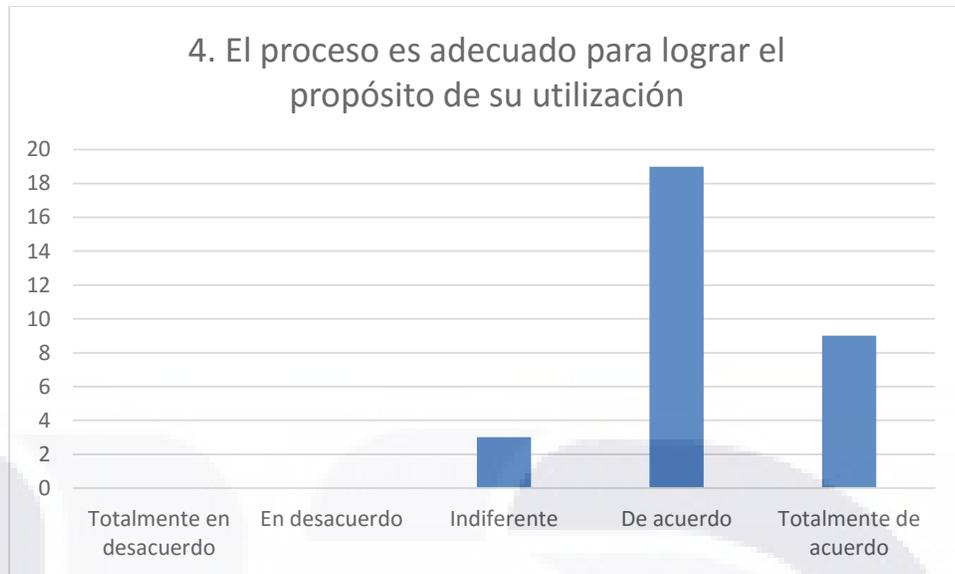
**Gráfica 25.** Segunda pregunta del instrumento de evaluación.

De manera similar a la sentencia anterior, 26 alumnos consideran que el marco teórico utilizado es relevante para el modelo planteado. Solo 4 participantes seleccionaron la opción "Indiferente" y solo 1 optó por la opción "En desacuerdo".



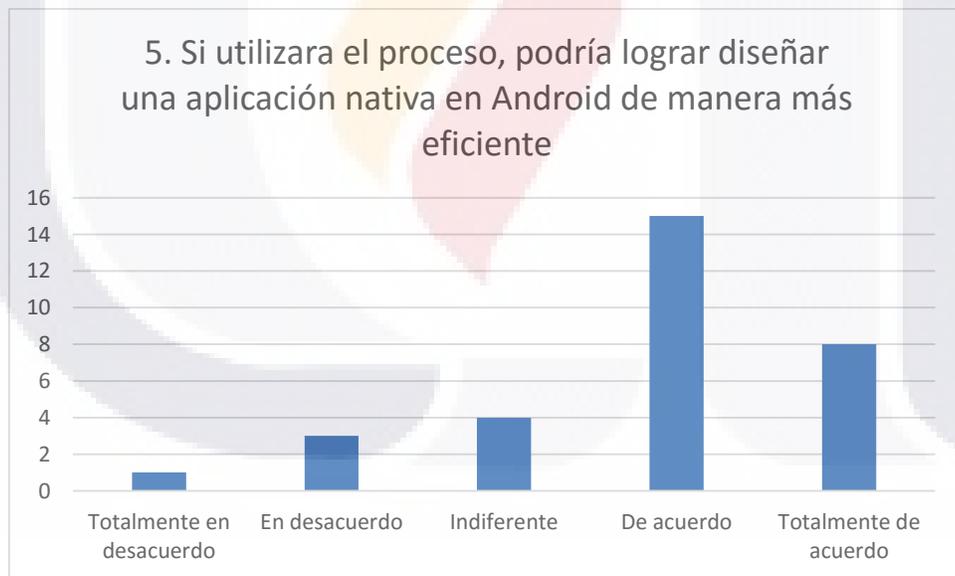
**Gráfica 26.** Tercera pregunta del instrumento de evaluación.

24 participantes mostraron su aprobación al indicar que en efecto, cada una de las etapas es coherente con el tema principal de la propuesta. Por su parte, 6 participantes seleccionaron la opción "Indiferente" y solo 1 se mostró "En desacuerdo".



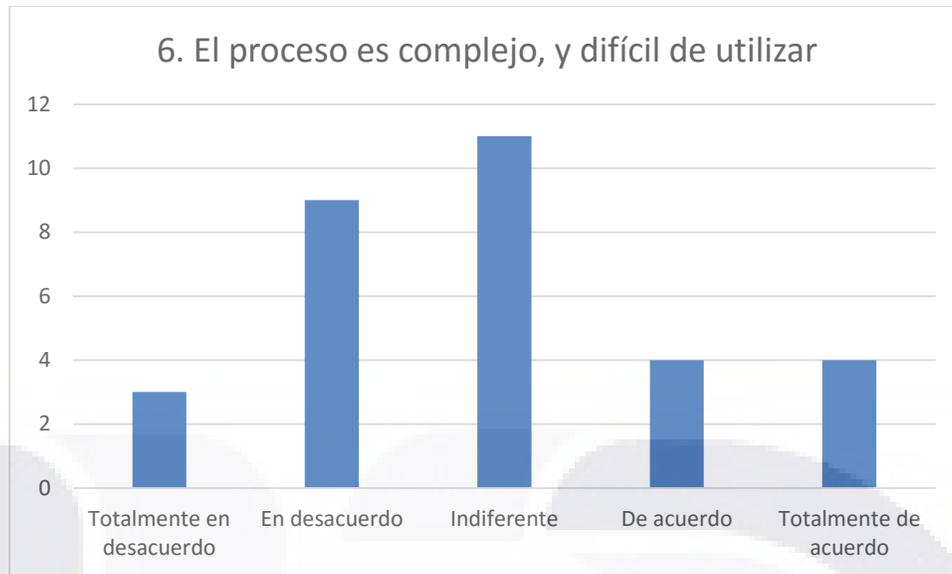
**Gráfica 27.** Cuarta pregunta del instrumento de evaluación.

La mayoría de los participantes expresaron su aprobación al indicar que el modelo es adecuado a su propósito, ya que 19 seleccionaron la opción "De acuerdo" y 9 "Totalmente de acuerdo". Por otro lado, solo 3 se mostraron indiferentes.



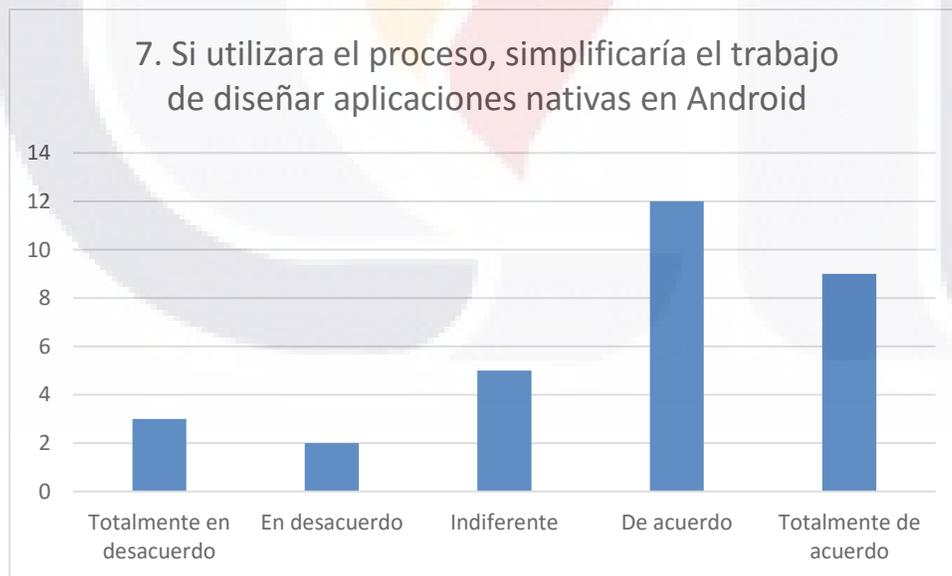
**Gráfica 28.** Quinta pregunta del instrumento de evaluación.

23 de los estudiantes consideran que de utilizar el proceso, podrían lograr diseñar una aplicación nativa en Android de manera más eficiente. Solo 4 estudiantes se mostraron indiferentes, 3 "En desacuerdo" y solo 1 seleccionó "Totalmente en desacuerdo".



**Gráfica 29.** Sexta pregunta del instrumento de evaluación.

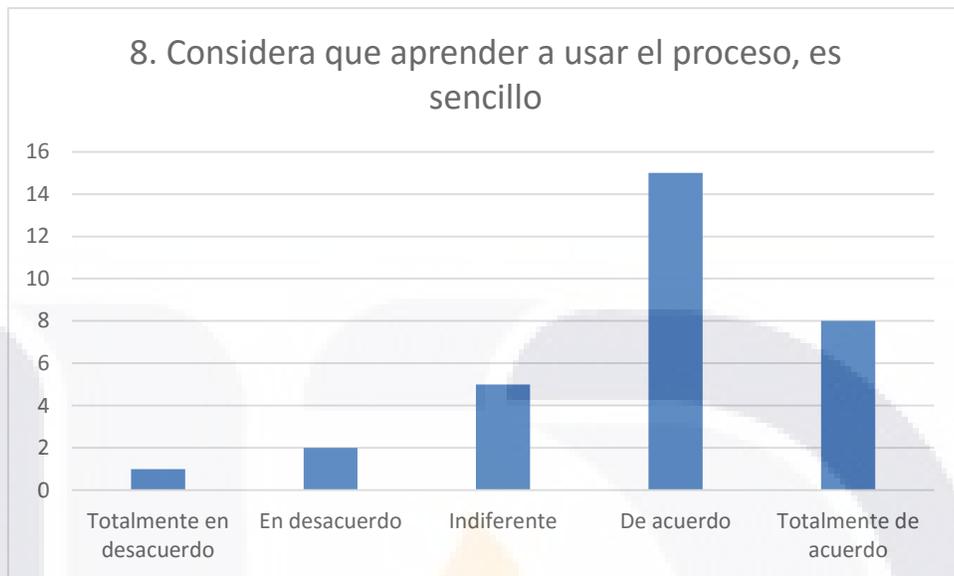
Con respecto a esta sentencia, sobre la complejidad y dificultad que representa utilizar el modelo se tienen opiniones divididas. Es decir, 11 participantes seleccionaron la opción "Indiferente", 8 están de acuerdo con la dificultad que implica utilizar el modelo, mientras que 12 que se mostraron en desacuerdo y consideran que no hay complejidad ni dificultad en absoluto.



**Gráfica 30.** Séptima pregunta del instrumento de evaluación.

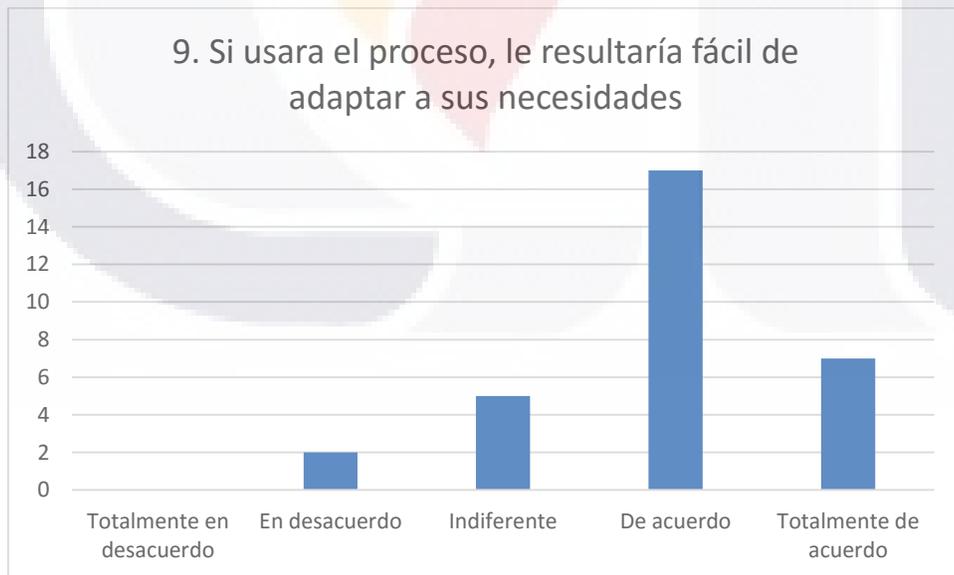
A diferencia de la sentencia anterior, 21 de los participantes están de acuerdo que de utilizar el modelo se simplificaría el trabajo de diseñar aplicaciones nativas en Android.

5 participantes se mostraron indiferentes, y solo 5 seleccionaron la opción "En desacuerdo".



**Gráfica 31.** Octava pregunta del instrumento de evaluación.

23 de los participantes indicaron estar de acuerdo con que aprender a usar el modelo es sencillo. Solo 5 participantes seleccionaron la opción "Indiferente", 2 más están "En desacuerdo", y solo uno "Totalmente en desacuerdo".



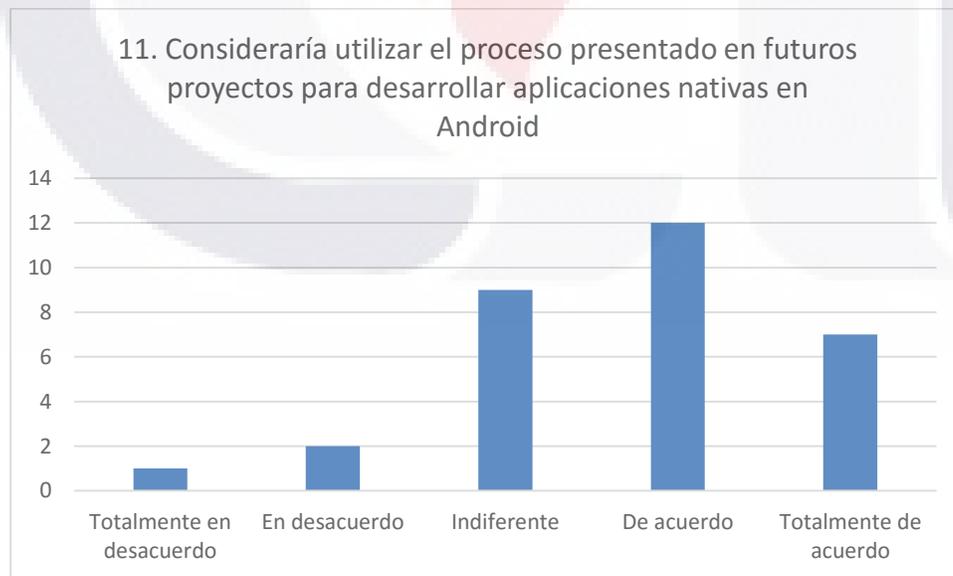
**Gráfica 32.** Novena pregunta del instrumento de evaluación.

24 de los participantes consideran que les resultaría fácil adaptar el modelo a las necesidades de sus proyectos. Solo 5 participantes seleccionaron la opción "Indiferente", y 2 se mostraron en desacuerdo.



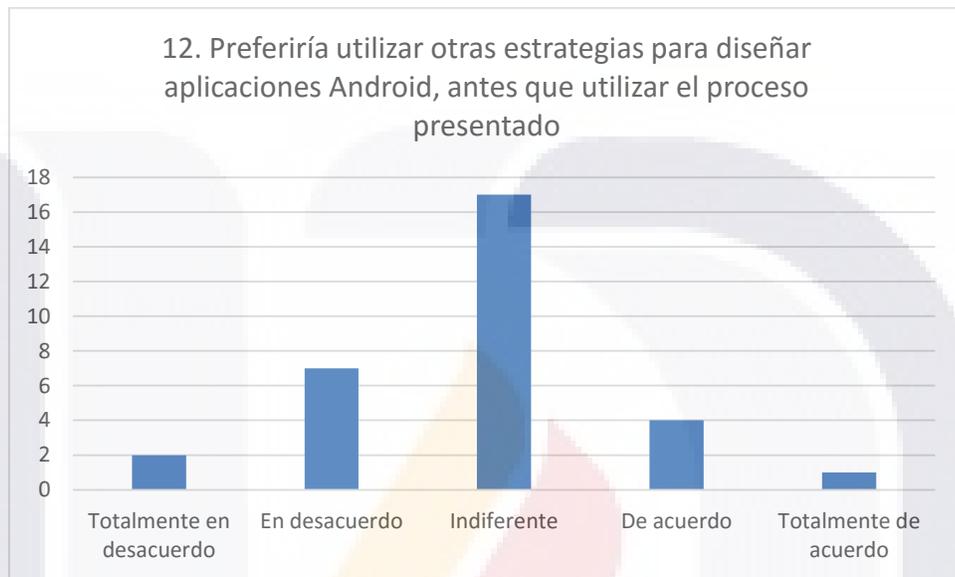
**Gráfica 33.** Décima pregunta del instrumento de evaluación.

Con respecto a la dificultad de poner en práctica el modelo, 16 de los participantes consideraron que no sería difícil, mientras que 5 si lo consideran una dificultad. 10 de los estudiantes seleccionaron la opción "Indiferente".



**Gráfica 34.** Décimo primer pregunta del instrumento de evaluación.

Gráfica 34. Pregunta 11. La mayoría de los participantes están de acuerdo con la idea de utilizar el modelo en futuros proyectos, ya que 19 se mostraron de acuerdo con ello. Por otra parte, 9 seleccionaron la opción "Indiferente", y únicamente 3 están en desacuerdo.



**Gráfica 35.** Décimo segunda pregunta del instrumento de evaluación.

17 de los participantes se mostraron indiferentes sobre utilizar otra estrategia para el desarrollo de aplicaciones en Android. 9 estudiantes optarían por implementar el modelo para el diseño de aplicaciones, y solamente 5 utilizarían otras estrategias.

## 9. Discusión de resultados

Antes de proceder al análisis individual del concepto, usabilidad, y aceptación del modelo propuesto, es necesario señalar los objetivos y criterios de evaluación considerados.

Los objetivos del caso de estudio son los siguientes:

- Medir empíricamente el concepto del modelo de desarrollo de aplicaciones nativas en Android basado en mejores prácticas y metodologías ágiles.
- Medir la percepción de los estudiantes sobre el modelo diseñado en cuanto a usabilidad y adopción.

Así también el criterio de evaluación seleccionado consiste en que el 80% de los participantes deben obtener al menos el 80% de los puntos en cada una de las categorías mencionadas. En la Tabla 6, se presentan los criterios utilizados para medir cada uno de los objetivos planteados.

- **Tabla 7.** Métricas de medición.

Objetivo de medición	Porcentaje	Equivalencia
<b>Concepto</b>	80%	12.8 puntos
<b>Usabilidad</b>	80%	19.2 puntos
<b>Aceptación</b>	80%	6.4 puntos

Es importante recordar que se consideran como resultados positivos la selección de cualquiera de las opciones "De acuerdo" y "Totalmente de acuerdo". Seleccionar "Indiferente" es de carácter neutral, y finalmente seleccionar cualquiera de las opciones "Totalmente en desacuerdo" y "En desacuerdo" son resultados negativos. Solamente en las sentencias 6, 10 y 12 la ponderación es a la inversa, es decir, seleccionar opciones *negativas*, es en realidad positivo para los fines del caso de estudio.

Para calcular el porcentaje de aceptación se suman los resultados positivos y se divide entre el total de participantes.

Las primeras cuatro sentencias del cuestionario buscan medir el concepto de modelo propuesto, con base a las métricas definidas el resultado es de 79.25%. Cabe mencio-

TESIS TESIS TESIS TESIS TESIS

nar que en las sentencias 1 y 3 el 23% y 19% respectivamente de los participantes se mostraron indiferentes.

Las sentencias de la 5 a la 10 tienen por objetivo medir la usabilidad del modelo donde se obtuvo un 63.5% de aceptación. Para esta sección, en promedio el 16% de la población se mostró indiferente, siendo en la sentencias 6 con un 35% y la sentencia 10 con 32% en donde hubo un mayor porcentaje de participantes que se mostraron indiferentes al respecto.

Finalmente las últimas dos sentencias se plantearon para medir la aceptación del modelo propuesto, obteniendo los siguientes resultados:

- La sentencia 11 obtuvo un 61% de aceptación y un 29% de indiferencia
- La sentencia 12 obtuvo un 29% de aceptación y un 54% de indiferencia

Por lo que el porcentaje de aceptación sobre la adopción del modelo es del 45%. Cabe señalar que del instrumento de evaluación, la última sentencia fue donde se obtuvo un mayor número de participantes que se mostraron indiferentes.

A primera vista los resultados parecen indicar que la hipótesis H0 es válida, la cual señala "No hay un conjunto de mejores prácticas, principios de metodologías ágiles, y elementos del área de interacción humano-computadora que constituyan una base para la construcción de un proceso de desarrollo de aplicaciones nativas para la plataforma Android."

Sin embargo, el análisis individual de cada una de las sentencias muestra que los resultados positivos fueron mayores en cada una de las preguntas que los resultados negativos. Para la población donde se aplicó el estudio el modelo propuesto representa una alternativa integral para el desarrollo de aplicaciones nativas en Android con respecto a otras opciones existentes.

Bajo esta primicia se considera que la hipótesis H1 la cual señala "Existe un conjunto no exhaustivo de mejores prácticas, principios de metodologías ágiles, y elementos del área de interacción humano-computadora que constituyen una base para la construcción de un proceso de desarrollo de aplicaciones nativas para la plataforma Android" es válida.



## 10. Conclusión

Al momento de concluir esta tesis, no fue posible encontrar un modelo o proceso similar para comparar y determinar las áreas de oportunidad.

Con respecto a los objetivos planteados al inicio de esta investigación es posible concluir, que el objetivo general, el cual es "definir un proceso de desarrollo para la construcción de aplicaciones nativas en Android basado en elementos de mejores prácticas, metodologías ágiles, e interacción humano-computadora para la plataforma Android" fue logrado satisfactoriamente.

Se revisó en la literatura la clasificación de las aplicaciones en Android y las diferentes estrategias disponibles para su desarrollo, indicado los beneficios y limitantes de cada una de ellas. Con base a la información recabada fue posible desarrollar y publicar una aplicación para tratamiento de la obesidad y la medición del estado de ánimo, todo esto bajo un enfoque empírico, con la información disponible en la web de desarrolladores de Google.

Posteriormente, fue posible evaluar las mejores prácticas detectadas en el proceso inicial, detectar las áreas de mejora, para generar una segunda versión de estas y finalmente incluirlas dentro del modelo a diseñar.

Con la información recolectada, las áreas y aspectos que se tomarían en cuenta de mejores prácticas, metodologías ágiles, e interacción humano – computadora, fue posible incorporar estos elementos dentro de un proceso de desarrollo de software para el diseño de aplicaciones nativas en Android. Lo anterior permitió dar cumplimiento a cada uno de los objetivos específicos planteados al inicio del proyecto.

Los resultados finales de la investigación sugieren que se debe aplicar nuevamente un caso de estudio a una población mayor de profesionales del área de Tecnologías de la Información, con la finalidad de comparar con los resultados obtenidos. Así también, el modelo propuesto puede comenzar a utilizarse en la práctica con el objetivo de seguir cada una de las fases, y determinar que efectivamente permiten desarrollar una aplicación en Android cumpliendo con los requerimientos y estándares de calidad seleccionados al inicio.

El ecosistema de dispositivos móviles crece a un ritmo acelerado, durante esta investigación se exploraron diferentes estrategias para el desarrollo de aplicaciones para la plataforma Android, y en cada una de estas ramas fue posible hacer contribuciones puntuales como la clasificación de aplicaciones en Android, describir un conjunto de características esenciales para aplicaciones de e-health, así como plantillas para el diseño de las mismas.

Como trabajo futuro se propone la implementación del modelo para el desarrollo de aplicaciones nativas en Android, así como un segundo caso de estudio que permitan generar un resultado definitivo.



## Glosario

Android	Sistema operativo de código abierto basado en Linux desarrollado por Open Handset Alliance.
Aplicación	Programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos
API	Application Programming Interface, es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
Framework	Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.
Interfaz de usuario	Medio con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo.
IDE	Integrated Development Environment, es una aplicación informática que proporciona servicios integrales para facilitar al desarrollador o programador el desarrollo de software.
Kernel	Software que constituye una parte fundamental del sistema operativo, y se define como la parte que se ejecuta en modo privilegiado (conocido también como modo núcleo)
POO	Paradigma de programación que usa objetos en sus interacciones, para diseñar aplicaciones y programas informáticos.
SDK	Software Development Kit, conjunto de herramientas de desarrollo de software que le permite al programador o desarrollador de software crear aplicaciones para un sistema concreto.
Smartphone	Tipo de teléfono móvil construido sobre una plataforma informática móvil, con mayor capacidad de almacenar datos y realizar actividades, semejante a la de una minicomputadora, y con una mayor conectividad que un teléfono móvil convencional.
Tablet	Computadora portátil de mayor tamaño que un teléfono inteligente

XML

te, integrada en una pantalla táctil con la que se interactúa primariamente con los dedos, sin necesidad de teclado físico ni ratón.  
eXtensible Markup Language, lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.



## Bibliografía

- [1].Tonini, A. R., Fischer, L. M., de Mattos, J. C. B., & de Brisolara, L. B. (2013, November). Analysis and evaluation of the Android best practices impact on the efficiency of mobile applications. In 2013 III Brazilian Symposium on Computing Systems Engineering (SBESC) (pp. 157-158). IEEE.
- [2].Oh, T., Stackpole, B., Cummins, E., Gonzalez, C., Ramachandran, R., & Lim, S. (2012, June). Best security practices for android, blackberry, and iOS. In Enabling Technologies for Smartphone and Internet of Things (ETSIoT), 2012 First IEEE Workshop on (pp. 42-47). IEEE.
- [3].Thung, P. L., Ng, C. J., Thung, S. J., & Sulaiman, S. (2010, June). Improving a web application using design patterns: A case study. In Information Technology (ITSim), 2010 International Symposium in (Vol. 1, pp. 1-6). IEEE.
- [4].Corral, L., Sillitti, A., & Succi, G. (2013, May). Software development processes for mobile systems: Is agile really taking over the business? In Engineering of Mobile-Enabled Systems (MOBS), 2013 1st International Workshop on the (pp. 19-24). IEEE.
- [5].Wasserman, A. I. (2010, November). Software engineering issues for mobile application development. In Proceedings of the FSE/SDP workshop on Future of software engineering research (pp. 397-400). ACM.
- [6].Firtman, M. (2010). Programming the mobile web. "O'Reilly Media, Inc."
- [7].Kraemer, F. A. (2011). Engineering android applications based on UML activities. In Model Driven Engineering Languages and Systems (pp. 183-197). Springer Berlin Heidelberg.
- [8].Vique, R. R. Métodos para el desarrollo de aplicaciones móviles. PID\_00176755.
- [9].Angulo, Roberto. (2013). Aplicaciones móviles híbridas: lo mejor de dos mundos. Debates IESA, 18(1), 80-81.
- [10]. Rainie, L. (2010). Internet, broadband, and cell phone statistics. Pew Internet & American Life Project, 5.
- [11]. Venta, L., Isomursu, M., Ahtinen, A., & Ramiah, S. (2008, September). "My phone is a part of my soul"—How People Bond with Their Mobile Phones. In Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM'08. The Second International Conference on (pp. 311-317). IEEE.
- [12]. Veloz Vidal, Carlos Alberto; Bustos Amador, Viviana; Álvarez Rodríguez, Francisco Javier; Mendoza González, Ricardo (2014). Identificación de las características esenciales para aplicaciones android de e-health orientadas al monitoreo de la obesidad. In Research in computer science (77). p 77-86.
- [13]. Hesse, B. W., & Shneiderman, B. (2007). eHealth research from the user's perspective. American journal of preventive medicine, 32(5), S97-S103.
- [14]. Norris, A. C., Stockdale, R. S., & Sharma, S. (2009). A strategic approach to m-health. Health informatics journal, 15(3), 244-253.
- [15]. Mendoza-González, R., Rodríguez, F. J. Á., Arteaga, J. M., & Mendoza-González, A. (2012, October). Guidelines for designing graphical user interfaces of mobile e-health communities. In Proceedings of the 13th International Conference on Interacción Persona-Ordenador (p. 3). ACM.

- TESIS TESIS TESIS TESIS TESIS
- [16]. Withers D.H.: Software engineering best practices applied to the modeling process. In: Simulation Conference Pro-ceedings, 2000. Winter, pp. 432-439 vol.1. (2000).
- [17]. Perks, M. (2003). Best practices for software development projects. Computerworld. Recuperado de: [http://www.ibm.com/developerworks/websphere/library/techarticles/0306\\_perks/perks2.html](http://www.ibm.com/developerworks/websphere/library/techarticles/0306_perks/perks2.html)
- [18]. Infotec, AMITI. 2013. "Mejores prácticas para la difusión del desarrollo y uso de aplicaciones móviles". Recuperado de: [http://amiti.org.mx/wp-content/uploads/2013/10/Estudio-Apps\\_Mejores-Pr%C3%A1cticas.pdf](http://amiti.org.mx/wp-content/uploads/2013/10/Estudio-Apps_Mejores-Pr%C3%A1cticas.pdf)
- [19]. Robillard, M. P. (2009). What makes APIs hard to learn? Answers from developers. *Software, IEEE*, 26(6), 27-34.
- [20]. Zuhud, D. A. Z., Abdul Rahman, N., & Ismail, M. (2013, March). A preliminary analysis on the shift of programming paradigms. In *Information and Communication Technology for the Muslim World (ICT4M), 2013 5th International Conference on* (pp. 1-5). IEEE.
- [21]. White, G., & Sivitanides, M. (2005). Cognitive differences between procedural programming and object oriented programming. *Information Technology and management*, 6(4), 333-350.
- [22]. Jackson, Wallace (2012). Chapter 5: Android Framework Overview. En (Second Edition) *Android Apps for Absolute Beginners* (pp. 99-124). New York. Editorial Apress.
- [23]. Veloz Vidal, Carlos Alberto; Álvarez Rodríguez, Francisco Javier; Mendoza González, Ricardo; Vargas Martín, Miguel (2014). Mejores prácticas para el desarrollo de aplicaciones nativas en Android de e-health y su publicación en la plataforma de Google Play. XXVII Congreso Nacional y XIII Congreso Internacional de Informática y Computación ANIEI.
- [24]. Bergmann, A., Li, J., Wang, L., Schulze, J., Bornstein, S. R., & Schwarz, P. E. (2007). A simplified Finnish diabetes risk score to predict type 2 diabetes risk and disease evolution in a German population. *Hormone and metabolic research= Hormon-und Stoffwechselforschung= Hormones et metabolisme*, 39(9), 677-682.
- [25]. Ordaz Lucas, Laura Yunuen; Gutiérrez Fernández, Yvonne; Veloz Vidal, Carlos Alberto; Álvarez Rodríguez, Francisco Javier (2014). Desarrollo de aplicaciones de e-health en Android para dispositivos móviles, diagnóstico para Diabetes Mellitus tipo II. XXVII Congreso Nacional y XIII Congreso Internacional de Informática y Computación ANIEI.
- [26]. Rodríguez, P. C., Guerrero, D. B., Díaz, S. P., & del Estudio, G. C. (2010). Elaboración y validación de un nuevo cuestionario de hábitos alimentarios para pacientes con sobrepeso y obesidad. *Endocrinología y nutrición*, 57(4), 130-139.
- [27]. Mariscal Pedraza, Omar; Solorio Mejía, María Elena; Veloz Vidal, Carlos Alberto; Álvarez Rodríguez, Francisco Javier (2014). Desarrollo de Aplicaciones Móviles para e-health: Aplicación Test de Obesidad. XXVII Congreso Nacional y XIII Congreso Internacional de Informática y Computación ANIEI 2014.
- [28]. Prochaska, J. O., & DiClemente, C. C. (1986). *Toward a comprehensive model of change* (pp. 3-27). Springer US.
- [29]. Saunders, J. B., Aasland, O. G., Babor, T. F., de la Fuente, J. R., & Grant, M. (1993). Development of the alcohol use disorders identification test (AUDIT): WHO
- TESIS TESIS TESIS TESIS TESIS

- collaborative project on early detection of persons with harmful alcohol consumption-II. *Addiction*, 88(6), 791-804.
- [30]. González-Forteza, C., Jiménez-Tapia, J. A., Ramos-Lira, L., & Wagner, F. A. (2008). Aplicación de la Escala de Depresión del Center of Epidemiological Studies en adolescentes de la Ciudad de México. *salud pública de méxico*, 50(4), 292-299.
- [31]. Web de Desarrolladores de Google. Recuperado de: <http://developer.android.com/develop/index.html>
- [32]. Brooks, F. P. (1975). *The mythical man-month* (Vol. 1995). Reading, MA: Addison-Wesley.
- [33]. Parada, A. G., & de Brisolará, L. B. (2012, November). A model driven approach for Android applications development. In *Computing System Engineering (SBESC), 2012 Brazilian Symposium on* (pp. 192-197). IEEE.
- [34]. Selic, B. (2006, April). Model-driven development: Its essence and opportunities. In *Object and Component-Oriented Real-Time Distributed Computing, 2006. ISORC 2006. Ninth IEEE International Symposium on* (pp. 7-pp). IEEE.
- [35]. PhoneGap. Recuperado de <http://phonegap.com/>
- [36]. Babu, R. M., Kumar, M. B., Manoharan, R., Somasundaram, M., & Karthikeyan, S. P. (2012). Portability of mobile applications using phonegap: A case study.
- [37]. Manifiesto Agil, Recuperado de <http://agilemanifesto.org/>
- [38]. Hossain, E., Babar, M. A., & Paik, H. Y. (2009, July). Using scrum in global software development: a systematic literature review. In *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on* (pp. 175-184). IEEE.
- [39]. Scharff, C. (2011, May). Guiding global software development projects using Scrum and Agile with quality assurance. In *Software Engineering Education and Training (CSEE&T), 2011 24th IEEE-CS Conference on* (pp. 274-283). IEEE.
- [40]. Apple. Recuperado de: <https://www.apple.com/ca/ipad-air-2/specs/>
- [41]. Selfa, D. M., Carrillo, M., & Del Rocío Boone, M. (2006, February). A Database and Web Application Based on MVC Architecture. In *Electronics, Communications and Computers, 2006. CONIELECOMP 2006. 16th International Conference on* (pp. 48-48). IEEE.
- [42]. GitHub, Recuperado de: <https://github.com/>
- [43]. Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012, February). Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1277-1286). ACM.
- [44]. Design Patterns, Recuperado de: <http://developer.android.com/design/patterns/index.html>
- [45]. Kaasila, J., Ferreira, D., Kostakos, V., & Ojala, T. (2012, December). Testdroid: automated remote UI testing on Android. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia* (p. 28). ACM.
- [46]. Williams, L. (2007). Lessons learned from seven years of pair programming at North Carolina State University. *ACM SIGCSE Bulletin*, 39(4), 79-83.
- [47]. Johnson, R. E., & Foote, B. (1988). Designing reusable classes. *Journal of object-oriented programming*, 1(2), 22-35.
- [48]. Hu, C., & Neamtiu, I. (2011, May). Automating GUI testing for Android applications. In *Proceedings of the 6th International Workshop on Automation of Software Test* (pp. 77-83). ACM.

- [49]. Android Testing. <http://developer.android.com/training/activity-testing/index.html>
- [50]. Memon, A. M. (2007). An event-flow model of GUI-based applications for testing. *Software Testing, Verification and Reliability*, 17(3), 137-157.
- [51]. Amalfitano, D., Fasolino, A. R., Tramontana, P., De Carmine, S., & Memon, A. M. (2012, September). Using GUI ripping for automated testing of Android applications. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering* (pp. 258-261). ACM.
- [52]. Serfass, D., & Yoshigoe, K. (2012, March). Wireless sensor networks using android virtual devices and near field communication peer-to-peer emulation. In *Southeastcon, 2012 Proceedings of IEEE* (pp. 1-6). IEEE.
- [53]. Chen, E. Y., & Itoh, M. (2010, June). Virtual smartphone over IP. In *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a* (pp. 1-6). IEEE.
- [54]. Vodafone. Recuperado de: <http://developer.vodafone.com/develop-apps/test/>
- [55]. Testdroid. Recuperado de: <http://testdroid.com/>
- [56]. Keynote. Recuperado de: <http://www.keynote.com/solutions/testing/mobile-testing>
- [57]. Políticas de publicación Google Play. <http://developer.android.com/distribute/tools/launch-checklist.html#understand-policies>



## Anexos

- A. Carlos Alberto Veloz Vidal, Francisco Javier Álvarez Rodríguez, Ricardo Mendoza González, Miguel Vargas Martin. (XXVII Congreso Nacional y XIII Congreso Internacional de Informática y Computación ANIEI 2014). Mejores prácticas para el desarrollo de aplicaciones nativas en Android de e-health y su publicación en la plataforma de Google Play.
- B. Laura Yunuen Ordaz Lucas, Yvonne Gutiérrez Fernández, Carlos Alberto Veloz Vidal, Francisco Javier Álvarez Rodríguez. (XXVII Congreso Nacional y XIII Congreso Internacional de Informática y Computación ANIEI 2014). Desarrollo de aplicaciones de e-health en Android para dispositivos móviles, diagnóstico para Diabetes Mellitus tipo II.
- C. Omar Mariscal Pedraza, María Elena Solorio Mejía, Carlos Alberto Veloz Vidal, Francisco Javier Álvarez Rodríguez. (XXVII Congreso Nacional y XIII Congreso Internacional de Informática y Computación ANIEI 2014). Desarrollo de Aplicaciones Móviles para e-health: Aplicación Test de Obesidad.
- D. Carlos Alberto Veloz Vidal, Viviana Bustos Amador, Francisco Javier Álvarez Rodríguez, Ricardo Mendoza González (2014). Identificación de las características esenciales para aplicaciones android de e-health orientadas al monitoreo de la obesidad. *Advances in Artificial Intelligence and its Intelligent Applications. Research in computer science* (77). p 77-86.
- E. Cuestionario utilizado en el caso de estudio. Carlos Alberto Veloz Vidal, Francisco Javier Álvarez Rodríguez, Miguel Vargas Martin, Ricardo Mendoza González.

## Mejores prácticas para el desarrollo de aplicaciones nativas en Android de e-health y su publicación en la plataforma de Google Play

Carlos Alberto Veloz Vidal <sup>1</sup>, Francisco Javier Álvarez Rodríguez <sup>1</sup>,  
Ricardo Mendoza González <sup>2,3</sup>, Miguel Vargas Martín <sup>4</sup>

<sup>1</sup> Universidad Autónoma de Aguascalientes, Centro de Ciencias Básicas, Dpto. Sistemas de Información Av. Universidad #940, Cd. Universitaria, CP. 20131, Aguascalientes, México.

– [carlosa.veloz@gmail.com](mailto:carlosa.veloz@gmail.com), [fjalvar@correo.uaa.mx](mailto:fjalvar@correo.uaa.mx)

<sup>2</sup> Instituto Tecnológico de Aguascalientes, Departamento de Sistemas y Computación, Av. Adolfo López Mateos 1801 Ote, Fracc. Bonagens, CP. 20257, Aguascalientes México

– [mendozagric@mail.ita.mx](mailto:mendozagric@mail.ita.mx)

– <sup>3</sup> Universidad Politécnica de Aguascalientes, Dirección de Posgrado e Investigación, Calle Paseo San Gerardo No. 207. Fracc. San Gerardo. C.P. 20342. Aguascalientes, México

– [ricardo.mendoza@upa.edu.mx](mailto:ricardo.mendoza@upa.edu.mx)

– <sup>4</sup> University of Ontario Institute of Technology, 2000 Simcoe Street N. Oshawa, Canada

– [miguel.vargasmartin@uoit.ca](mailto:miguel.vargasmartin@uoit.ca)

**Resumen.** La popularidad de los dispositivos móviles que operan con Android ha aumentado en los últimos años, propiciando que los desarrolladores generen diversas herramientas y aplicaciones (apps) para todo el público. Existen diferentes recursos disponibles para el desarrollo y publicación de apps, incluidas mejores prácticas, sin embargo, la mayoría de esta información no está orientada a programadores que recién comienzan a iniciarse en el desarrollo de apps en Android. Los inconvenientes más evidentes, involucran el desconocimiento de los componentes de una app y de las herramientas disponibles, así como la falta de una guía que oriente a los desarrolladores durante el proceso de desarrollo-publicación de la app. Con el objetivo de reducir esta brecha, se propone una colección no exhaustiva de mejores prácticas para el desarrollo de apps nativas en Android y su publicación en la plataforma Google Play.

**Palabras clave:** dispositivos móviles, aplicación Android, e-health, mejores prácticas para el desarrollo de software.

## 1 Introducción

Android opera en millones de dispositivos móviles en más de 190 países de todo el mundo. Es la mayor base instalada de cualquier plataforma móvil y de mayor crecimiento, cada día un millón de usuarios encienden sus dispositivos Android por primera vez y comienzan a buscar aplicaciones (apps), juegos, libros electrónicos y demás contenido digital [1].

Las apps en Android se basan en eventos y constantemente responden a cualquier entrada que se detecte a través de las interfaces de usuario y/o sensores con los que cuenta el dispositivo móvil [2]. También, son relativamente sencillas de desarrollar, ya que se crean utilizando el lenguaje Java, el cual pertenece al paradigma de Programación Orientada a Objetos (POO), por lo que cualquier desarrollador que conozca este paradigma puede crear y publicar apps. En línea se encuentran una gran variedad de ambientes de desarrollo (SDK, por sus siglas en inglés), además, de la web de desarrollo de Android que provee documentación, códigos de muestra y una sección dedicada a las buenas prácticas para el diseño, desarrollo, seguridad, pruebas y ejecución a tener en cuenta para la creación de una app en Android [3].

Existen diferentes tipos de aplicaciones móviles, independientes del dispositivo donde se instalaran y ejecutaran (televisores inteligentes, celulares, tablets, relojes, etc.), esto debido a la facilidad para implementar y desarrollar apps que brinda este sistema operativo. A continuación, se muestra una clasificación de las aplicaciones en Android de acuerdo a su funcionalidad [4 y 5].

1) *Aplicaciones básicas*, son aplicaciones simples de interacción básica, las cuales solamente envían o reciben información. Un ejemplo de esta categoría son las aplicaciones para mensajes SMS. 2) *Aplicaciones Web o web móviles*, similares a las páginas web que se pueden acceder desde un equipo de cómputo convencional, son fáciles y económicas de desarrollar ya que utilizan tecnología HTML, CSS, entre otras, como cualquier página de internet. Un inconveniente de este tipo de aplicaciones es que no pueden inter-actuar con los sensores propios de un dispositivo móvil, como lo es la cámara, el GPS, acelerómetro, etc. 3) *Aplicaciones nativas*, aplicaciones propias de cada plataforma. En este caso, son las aplicaciones que se desarrollan con el uso del SDK oficial de Android, para explotar al máximo las características de este sistema operativo. 4) *Aplicaciones híbridas*, reúnen las mejores características de las aplicaciones web y las aplicaciones nativas. Una aplicación híbrida se basa en el desarrollo de una página móvil con capacidad para manejar los elementos nativos del dispositivo (cámara y GPS, entre otros sensores).

Por otro lado, e-health se puede definir como “El empleo de información y tecnologías de comunicación para un mejor control de la salud. Por ejemplo, para el tratamiento de determinados pacientes, fomentar la investigación, crear herramientas para la educación de estudiantes, hacer detección de diversas enfermedades, y en fin, para la supervisión de la salud pública”. Esto de acuerdo a la Organización Mundial de la Salud (OMS).

Lo mencionado anteriormente, favorece a que los desarrolladores puedan crear aplicaciones para fomentar la cultura de la concientización de diversos padecimientos entre los usuarios, estos productos pueden llegar a ser publicados en plataformas como Google Play, la cual es el mecanismo oficial de distribución detrás de las apps Android. Sin

embargo, se requiere del uso de mejores prácticas ya que, “una mejor práctica puede ser una práctica técnica o de gestión, que ha demostrado una mejora sobre uno o más aspectos como: productividad, coste, calendario, calidad o satisfacción de usuario” [6]. De esta manera se asegura que el producto final cuente con la calidad necesaria en términos de la Ingeniería de Software, y al mismo tiempo sea de utilidad para los usuarios que lo descarguen.

En la literatura existen varios trabajos que hablan sobre las mejores prácticas, pero la mayoría de ellos se enfocan en aspectos relacionados al desempeño y consumo de energía, seguridad e integridad de la información, tales son los casos respectivamente de [7 y 8]. En México y Latinoamérica se han publicado mejores prácticas, que efectivamente sirven para fomentar el desarrollo de apps y su consumo dentro de los distintos niveles de la sociedad [9]. Prácticas que son de utilidad, sin embargo, no resuelven del todo el problema de orientar al desarrollador durante el proceso de desarrollo y publicación de apps. Por otra parte, la documentación que se presenta en la web de desarrolladores de Google, la cual coincide con lo presentado en los libros como [10 y 11], describen en detalle cada uno de los elementos dentro de la Application Programming Interface (API), sus métodos, atributos y propiedades; incluso se presentan fragmentos de código para mostrar su implementación. Sin embargo, no representan una solución completa al problema del desarrollo y publicación de apps en Android, pues necesario complementar este proceso con el uso de mejores prácticas, de esta manera el desarrollador tendrá las herramientas para generar productos de calidad y utilidad para los usuarios finales.

Lo anterior representa la motivación principal para la realización de este trabajo, cuyas secciones posteriores están organizadas de la siguiente manera: La sección 2, presenta la brecha que se desea acortar, siguiéndole la propuesta de solución descrita en la sección 3. En la sección 4, se ejemplifica la aplicación de la propuesta con un caso de estudio en el que se muestra el proceso desarrollo-publicación de una app para el monitoreo de obesidad (disponible en [12]). Finalmente, los resultados del estudio se muestran en la sección 5, concluyendo los aspectos más relevantes en la sección 6, complementados con algunos caminos a explorar en el futuro cercano.

## **2 Problemática**

Como se mencionó anteriormente, es necesario complementar el material existente que describe las características y componentes de una aplicación en Android, con el uso de mejores prácticas orientadas al desarrollo y publicación de estos productos en plataformas como Google Play.

La documentación oficial para el desarrollo de apps en Android, así como algunos libros, describen en detalle los componentes de la API y sus características, presentan fragmentos de código que muestran su implementación dentro de un proyecto, si bien es información de utilidad para el desarrollador, no describen claramente el camino a seguir para comenzar a generar apps, ya que no proveen una imagen general de cómo interactúan los distintos componentes de una aplicación, además, los códigos muestra-

dos pudieran resultar muy sencillos y su nivel de complejidad no se compara con el que se pretende alcanzar en la mayoría de apps a desarrollar [14].

Un estudio de las mejores prácticas disponibles para aplicaciones Android permitió observar que, la mayoría de estas están orientadas a resolver problemas de seguridad, integridad y confidencialidad de la información sensible del usuario, riesgos y otros temas relacionados con la seguridad de la información, tal es el caso de [7 y 8]. Sin duda, esta información debe ser tomada en cuenta por el desarrollador al momento de generar aplicaciones, pero en lo relacionado con el tema de cómo generar y publicar una app, no es exactamente el contenido que el desarrollador necesita. De manera adicional, existen mejores prácticas que fomentan el desarrollo y consumo de apps en todos los niveles de la sociedad, en las cuales se describen estrategias que van orientadas hacia el punto de vista comercial [9].

Por lo tanto, las mejores prácticas que se proponen a continuación pueden servir a los desarrolladores como un complemento a la información disponible desde el punto de vista tecnológico, ya que se consideraron elementos de la Ingeniería de Software para aumentar la productividad, reducir el tiempo de desarrollo, generar un producto que guste al usuario y que al mismo tiempo sea útil.

### **3 Mejores prácticas para el desarrollo de aplicaciones nativas en Android de e-health y su publicación en la plataforma de Google Play**

Antes de presentar las mejores prácticas, es importante mencionar los componentes con los que cuenta el sistema operativo Android, el cual fue diseñado para proveer la mayor modularidad posible a los desarrolladores, lo que permite identificar de manera clara los cuatro componentes que conforman una app: 1) *Actividades*, manejan la interfaz del usuario que se muestra en la pantalla del dispositivo móvil; 2) *Servicios*, procesos que se ejecutan en paralelo a la aplicación; 3) *Receptores de transmisiones*, se encargan de la comunicación dentro de la app como también con otras aplicaciones; y 4) *Proveedores de contenido*, se encargan del manejo de los datos y la administración de operaciones relacionadas con bases de datos [11].

Las mejores prácticas para el desarrollo de apps nativas fueron elaboradas para utilizarse en el componente de *Actividades*, el cual es vital para una app ya que por medio de estas se presenta la interfaz gráfica al usuario a través de distintos tipos de archivos de diseño (Layouts), que utilizan eXtensible Markup Language (XML).

Existen varios tipos de apps móviles, independientes del dispositivo donde se instalaran y ejecutaran (Televisores inteligentes, celulares, tablets, relojes, etc.). Las mejores prácticas propuestas, están dirigidas a apps nativas, las cuales son aplicaciones propias de cada plataforma. En este caso, las apps nativas son aquellas que se desarrollan con el uso del SDK oficial, lo que permite explotar al máximo las características de este sistema y del dispositivo donde se ejecutaran [4 y 5].

Pensando en un proceso de Ingeniería de Software para la construcción de un producto, las mejores prácticas propuestas se presentan de la siguiente manera:

Fase del ciclo de desarrollo	Práctica propuesta	Descripción
<b>Requerimientos</b>	Selección de los dispositivos móviles	Analizar los requerimientos y el alcance de la app a construir, y así seleccionar el dispositivo móvil que mejor se adapte a estas necesidades (tablet o smartphone).
<b>Diseño</b>	Identificar las resoluciones de pantalla a soportar	Es indispensable aprovechar al máximo el espacio disponible en cada resolución de pantalla, por lo que es conveniente realizar un estudio de mercado y determinar las resoluciones de pantalla mas populares de acuerdo al dispositivo movil seleccionado.
<b>Desarrollo</b>	Elegir un tipo de archivo de diseño (layout)	Existen diferentes tipos de layouts en la API de Android, cada uno con propiedades y comportamientos unicos, por lo que es conveniente analizar cada uno de ellos y elegir aquel que mejor se adapta a los requerimientos de la app a diseñar.
<b>Liberación</b>	Llenado de la ficha de Google Play	Determinar si el producto es lo suficientemente maduro como para ser liberado en producción, alpha testing o beta testing, segun corresponda. Tambien llenar la información solicitada en cada apartado de la ficha de Google Play, ya que en base a su contenido el usuario decide descargar la app.

**Tabla 1.** Mejores prácticas propuestas para el desarrollo y publicación de una app Android.

Como se puede observar en la Tabla 1, el trabajo propuesto impacta directamente en casi todas las fases del ciclo de desarrollo de una aplicación, se busca que con su uso y adopción dentro de los desarrolladores, se reduzca el tiempo invertido en analizar la documentación oficial, pues con base en un análisis de los requerimientos y el alcance que se pretende lograr con la aplicación a generar, el desarrollador ya solo se referirá a la documentación oficial buscando información muy precisa para ciertos componentes.

#### 4 Caso de estudio

Para ejemplificar y probar cada una de las mejores prácticas propuestas, se desarrolló una aplicación para el Monitoreo de Obesidad, la cual está disponible en [13]. Actualmente México ocupa el primer lugar mundial en obesidad, lo que representa uno de los principales problemas de salud pública del país. Esto ha despertado el interés, y la necesidad de que la comunidad tecnológica y científica se unan a los especialistas de la salud, para generar estrategias que refuercen los tratamientos preventivos tradicionales para el control del peso. Una de estas estrategias es el desarrollo de aplicaciones e-health para el sistema operativo Android. Si bien, existen aplicaciones disponibles para este fin, la mayoría no cuentan con un respaldo de profesionales de la salud, haciendo de sus diagnósticos, y/o sugerencias poco confiables.

Dicha aplicación consta de 36 preguntas, las cuales se pueden clasificar en tres grupos que son: 1) Hábitos alimenticios, 2) Actividad física y 3) Estado de ánimo. Además, este cuestionario permite identificar de manera más precisa el estado de salud de un usuario ya que considera factores como el consumo de azúcar, alimentación saludable, ejercicio físico, contenido calórico, bienestar psicológico, tipo de alimentos, conocimiento y control y por último consumo de alcohol [15].

En la fase de requerimientos se aplicó la práctica propuesta para selección de dispositivos móviles, se determinó que la mejor manera para distribuir esta aplicación entre todos los sectores de la población era brindando soporte tanto a smartphones como a tablets y con esto conseguir una mayor distribución entre los usuarios.

Ya que se seleccionaron los dispositivos móviles, es importante identificar a que resoluciones de pantalla en específico se les dará soporte. Se utilizó la práctica propuesta para etapa de diseño. La cual nos permite identificar las resoluciones de pantalla a soportar; para esto fue necesario realizar un análisis de los tipos de smartphones y tablets más vendidos en el mercado, e identificar qué características técnicas comparten, con base a este estudio las resoluciones de pantalla a soportar fueron de 2.7, 4.1, 5.1 pulgadas para smartphones y 10.1 para tablets. Uno de los principales inconvenientes para desarrollar esta aplicación, es que el API de Android contiene diferentes tipos de layouts, cada uno con propiedades y comportamientos distintos. Tomando la práctica propuesta para la fase de desarrollo, se analizaron los distintos layouts refiriéndose a la documentación oficial para compararlos. Se optó por elegir el Absolute Layout, ya que permite definir la posición absoluta en coordenadas (x, y) de cada uno de los elementos de la interfaz de usuario. Al navegar entre cada una de las pantallas se mostraban los elementos en la misma posición, sin importar la resolución del dispositivo final, lo que agrega homogeneidad a la aplicación.

Finalmente, ya que se tiene la aplicación construida, probada y está lista para su distribución, se toma la práctica propuesta para la fase de liberación, donde se genera el archivo con extensión .apk (*Application Package File*), el cual es requerido por la ficha de Google Play al momento de liberar un nuevo producto. Se ingresa la información requerida por la consola de desarrolladores para los campos *Título*, con un nombre que resulte atractivo y fácil de recordar para los usuarios, *Descripción* donde se mencionan las características de esta app, se incluyen capturas de pantalla para que se pueda dar un vistazo de como luce la app antes de descargarse en smartphones y/o tablets, además, se agrega un icono para completar la ficha de Google Play. De manera adicional a esta información, se debe seleccionar la categoría y tipo de contenido de la aplicación, en este caso corresponde a aplicaciones del área de *Medicina*, con tipo de contenido *Para todos*. En cuanto al precio, como se mencionó anteriormente será una aplicación de descarga gratuita para beneficiar al mayor número de usuarios posible, por otro lado, la aplicación se distribuirá en todos los países donde Google Play tiene presencia, considerando que el único idioma al que se da soporte es el español.

## 5 Resultados

Los resultados que se presentan en esta sección, fueron obtenidos a través de la consola para desarrolladores de Google, la cual, permite dar un seguimiento adecuado de la aceptación y distribución de la app entre los usuarios.

La implementación de las prácticas se ve reflejada en la información general del archivo apk generado, como se muestra en la Figura 1. Donde se puede observar el número de dispositivos compatibles (6805), niveles de la API (8+), y un tamaño de la app pequeño (1.03 MB) en comparación con otras aplicaciones disponibles, lo cual facilita su distribución entre los usuarios

### INFORMACIÓN DEL APK

com.mipaquete.prototipo1

Código de versión	Nombre de la versión	Tamaño
<b>3</b>	<b>1.2</b>	<b>1,03 MB</b>
Dispositivos Android compatibles	<b>6805 dispositivos</b>	
Niveles de API	<b>8+</b>	
Diseños de pantalla	<b>4 diseños de pantalla</b>	
small		
normal		
large		
xlarge		

**Fig. 1.** Información del archivo .apk generado

La versión actual de la aplicación Monitoreo de Obesidad es la tercera que se libera, y fue publicada el 9 de Febrero de 2014, hasta el momento ha sido descargada un total de 555 ocasiones y se encuentra activa en 56 dispositivos, además de contar con una calificación de 4.17 estrellas de un máximo de 5, lo que indica que el producto generado ha resultado de utilidad para los usuarios que lo han descargado. Información obtenida al 29 de Agosto del 2014.

## 6 Conclusiones y trabajo futuro

Para este trabajo de investigación se analizaron las mejores prácticas para el desarrollo de aplicaciones presentadas en la documentación oficial de Google, así como también distintos trabajos que abarcan temas de seguridad y manejo de la información, hasta mejores prácticas para la producción y consumo de apps [3, 7, 8 y 9 respectivamente].

Si bien es información de utilidad para el desarrollador, no lo orienta correctamente durante el proceso de desarrollo y publicación, ya que las mejores prácticas que aquí se presentaron están orientadas a programadores con conocimientos básicos de POO y que recién comienzan a desarrollar apps móviles en Android. Además con apego a un pro-

ceso formal de Ingeniería de Software, las mejores prácticas propuestas influyen directamente en una etapa del ciclo de desarrollo, buscando reducir el tiempo de codificación, aumentar la productividad consultando solo la documentación relacionada con los componentes utilizados, permitiendo así generar un producto que tenga la calidad que el usuario busca en una app, y que al mismo tiempo de le sea de utilidad.

La app publicada para el Monitoreo de Obesidad (ver [13]), hasta el momento se encuentra catalogada en la categoría de descargas de 500 a 1000, y principalmente se ha descargado en países como Argentina, España, México y Estados Unidos, llegando a obtener varias calificaciones de cinco estrellas sin reportes de errores o fallos esto de acuerdo a lo reportado por la consola de desarrolladores de Google.

Como trabajo futuro se pretende extender el número de mejores prácticas, así como probar su efectividad en el desarrollo de aplicaciones de e-health para medir cuantitativa y cualitativamente su efectividad.

## Referencias

- [58]. Google. About Android. Recuperado de <http://developer.android.com/about/index.html>
- [59]. F. A. Kraemer, "Engineering android applications based on uml activities," Proceedings of the 14th international conference on Model driven engineering languages and systems (MODELS'11), Jon Whittle, Tony Clark, and Thomas Khne (Eds.). Springer-Verlag, Berlin, eidelberg, pp. 183–197, 2011.
- [60]. Google. Develop, Training. Recuperado de <https://developer.android.com/training/index.html>
- [61]. Robert Ramirez Vique. (2013). Características de un proyecto de desarrollo para dispositivos móviles En 1ra edición, Métodos para el desarrollo de aplicaciones móviles. Catalunya, España. Editorial UOC.
- [62]. Angulo, Roberto. (2013). Aplicaciones móviles híbridas: lo mejor de dos mundos. Debates IESA, 18(1), 80-81.
- [63]. Withers D.H.: Software engineering best practices applied to the modeling process. In: Simulation Conference Proceedings, 2000. Winter, pp. 432-439 vol.1. (2000).
- [64]. Aline Tonini, Leonardo Fischer, Júlio Mattos, Lisane Brisolará, "Analysis and evaluation of the Android best practices impact on the efficiency of mobile applications", 2013 III Brazilian Symposium on Computing Systems Engineering, vol., no., pp.157, 158, 4-8 Dec. 2013
- [65]. Tae Oh; Stackpole, B.; Cummins, E.; Gonzalez, C.; Ramachandran, R.; Shinyoung Lim, "Best security practices for android, blackberry, and iOS," Enabling Technologies for Smartphone and Internet of Things (ETSIoT), 2012 First IEEE Workshop on , vol., no., pp.42,47, 18-18 June 2012.
- [66]. Infotec, AMITI. "Mejores prácticas para la difusión del desarrollo y uso de aplicaciones móviles". Recuperado de: [http://amiti.org.mx/wp-content/uploads/2013/10/Estudio-Apps\\_Mejores-Pr%C3%A1cticas.pdf](http://amiti.org.mx/wp-content/uploads/2013/10/Estudio-Apps_Mejores-Pr%C3%A1cticas.pdf)
- [67]. Michael Burton, Donn Felker. (2013). Chapter 8: Publishing Your App to the Google Play Store. En (Second Edition) Android Application Development for Dummies (pp. 183-202). Hoboken, New Jersey. Editorial Wiley.
- [68]. Wallace Jackson (2012). Chapter 5: Android Framework Overview. En (Second Edition) Android Apps for Absolute Beginners (pp. 99-124). New York. Editorial Apress.

- [69]. Google. Monitoreo de Obesidad. Recuperado de <https://play.google.com/store/apps/details?id=com.mipaquete.prototipo1&hl=es>
- [70]. M. P. Robillard, "What makes APIs hard to learn? Answers from developers," IEEE Software, vol. 26, no. 6, pp. 26–34, 2009.
- [71]. Castro Rodríguez P, Bellido Guerrero D, Pertega Díaz S. Elaboración y validación de un nuevo cuestionario de hábitos alimentarios para pacientes con sobrepeso y obesidad. Endocrinología y Nutrición 2010; 57 (4): 130-139.



## Desarrollo de aplicaciones de e-health en Android para dispositivos móviles, diagnóstico para Diabetes Mellitus tipo II

<sup>1</sup>Laura Yunuen Ordaz Lucas, <sup>2</sup>Yvonne Gutiérrez Fernández, <sup>3</sup>Carlos Alberto Veloz Vidal, <sup>3</sup>Francisco Javier Álvarez Rodríguez

<sup>1</sup>Instituto Tecnológico de Jiquilpan, Facultad de Ingeniería en Sistemas Computacionales Carretera Nacional Kilómetro 202, Centro, C.P. 59510 Jiquilpan de Juárez, Michoacán, México.  
lyol901018@hotmail.com

<sup>2</sup>Universidad de Colima, Facultad de Ingeniería Mecatrónica, carretera Manzanillo-Cihuatlán km. 20, Ejido El Naranjo C.P. 28864, Manzanillo, Colima, México.  
yvogtz@gmail.com

<sup>3</sup>Universidad Autónoma de Aguascalientes, Centro de Ciencias Básicas, Dpto. Sistemas de Información Av. Universidad #940, Cd. Universitaria, C.P. 20131, Aguascalientes, México.  
carlosa.veloz@gmail.com, fjalvar@correo.uaa.mx

**Resumen.** En la actualidad la diabetes se encuentra entre las principales causas de muerte en México, lo que representa uno de los principales problemas de salud pública del país. Con la finalidad de crear conciencia sobre este padecimiento y complementar los tratamientos tradicionales, surge la necesidad de desarrollar aplicaciones de e-health para el sistema operativo Android. Existen varias aplicaciones disponibles para este fin, sin embargo, no cuentan con el respaldo de instituciones médicas y/o especialistas, por lo que los resultados, información, o diagnósticos resultan poco confiables. Para contribuir a reducir esta brecha, se desarrolló una aplicación que implementa el test Findrisk, para determinar si un usuario está en riesgo de contraer esta enfermedad en los próximos 10 años, además de incluir consejos para guiar a los usuarios hacia una mejor calidad de vida. Esta app se distribuye a través de Google Play y los resultados obtenidos se discuten al final del artículo.

**Palabras clave:** diabetes, e-health, aplicación Android, dispositivos móviles.

### Introducción

La diabetes mellitus o diabetes, es una enfermedad crónica degenerativa que se presenta cuando el páncreas no produce insulina, o bien, la que se produce no es utilizada de manera eficiente por el organismo; ésta es la hormona responsable de que la glucosa de los alimentos sea absorbida por las células y dota de energía al organismo [1].

La diabetes es un problema a nivel mundial, por que la incidencia, prevalencia y mortalidad están incrementándose a un ritmo acelerado. La diabetes se está mostrando en etapas de la vida cada vez más tempranas; en el caso de la tipo I, también conocida

como diabetes juvenil, la causa una acción autoinmune del sistema de defensa del cuerpo que ataca a las células que producen la insulina y se presenta en pacientes con antecedentes familiares directos. La tipo II, además del antecedente heredofamiliar, depende de estilos de vida como son el sobrepeso, dieta inadecuada, inactividad física, edad avanzada, hipertensión, etnicidad e intolerancia a la glucosa; además, en las mujeres se presenta en aquellas con antecedentes de diabetes gestacional y alimentación deficiente durante el embarazo [2, 3].

De acuerdo a la Organización Mundial de la Salud (OMS), e-health se define como “El empleo de información y tecnologías de comunicación para un mejor control de la salud. Por ejemplo, para el tratamiento de determinados pacientes, fomentar la investigación, crear herramientas para la educación de estudiantes, hacer detección de diversas enfermedades, y en fin, para la supervisión de la salud pública”.

El desarrollo de aplicaciones de e-health para el sistema operativo Android constituye una valiosa herramienta por su buen aprovechamiento de este tipo de apps, contando con el respaldo de una institución médica y/o especialistas de la salud. Además, como es el caso de la aplicación a desarrollar para el tratamiento de la diabetes Mellitus tipo II, se busca que los resultados obtenidos ayuden a difundir entre los usuarios la conciencia de la prevención de este tipo de padecimientos, en ningún momento los resultados, consejos, y/o información que se presentan al usuario podrán reemplazar la opinión de un especialista, al contrario, la idea es reforzar los tratamientos tradicionales para este tipo de padecimientos con el uso de los dispositivos móviles.

Lo anterior, representa la motivación principal para la realización de este trabajo, cuyas secciones posteriores se organizan de la siguiente manera: La sección 2, presenta la brecha que se desea acortar, siguiéndole la solución propuesta en la sección 3. En la sección 4, se muestran los resultados parciales de la aplicación distribuida a través de la plataforma Google Play. Finalmente las conclusiones y el trabajo futuro se presentan en la sección 5, así como el camino a seguir en el corto plazo.

## **Problemática**

En la actualidad las personas son más propensas a adquirir enfermedades, por lo cual existe la necesidad de atender su salud para la prevención o tratamiento de dichos padecimientos. Por lo que se ha puesto un especial interés en la enfermedad de la Diabetes Mellitus tipo II, ya que este padecimiento es de carácter internacional, no obstante este tipo de padecimiento, a veces, repercute de manera negativa en la salud física y mental, generando alteraciones en el estilo de vida de quienes la padecen.

Ante la búsqueda de soluciones a estos problemas, es entonces que nace la inquietud de realizar una aplicación en Android la cual puede ser una solución para cumplir dicho objetivo e incluso producir efectos a nivel internacional.

Por otro lado se aprovecha el avance de la tecnología, asimismo el auge con el que cuentan las aplicaciones gratuitas en Android, dado que es una demanda a la cual responde el hombre moderno. Cabe destacar que las apps existentes, de control y monitoreo de la Diabetes Mellitus tipo II son de nivel de madurez bajo y no tratan de manera

sería el problema de la Diabetes, además, de que la gran mayoría no cuenta con el respaldo por parte de alguna institución médica y/o especialistas de la Diabetes, por lo que representan sin duda un problema, ya que brindan información poco confiable, lo cual puede liberar inconvenientes adicionales a la salud. Si bien es cierto desarrollar e implementar una aplicación en Android cualquiera la puede realizar, pero a través de un largo estudio, se obtuvo que la gran mayoría de las aplicaciones dentro de la categoría de Medicina no poseen las características y requerimientos necesarios para ser valorada como una aplicación eficiente.

El análisis realizado, permitió identificar varios aspectos de mejora que pueden integrarse en las siguientes características básicas para una app de e-health, las cuales son: 1) *Tamaño*, no debe pesar demasiados Mb ya que no resultará atractivo para un usuario descargar una app “pesada”, 2) *Exenta de publicidad*, las apps de e-health no deben contener publicidad, ni cualquier otro distractor que desvíe la atención del objetivo original de la app, 3) *Respaldo científico*, al tratar temas delicados de salud, debe de haber un estudio científico por profesionales en el área, que permita generar un diagnóstico adecuado a la condición de cada usuario; esto se debe indicar en la ficha de Google Play, 4) *Ser gratuita*, ya que el objetivo principal de este tipo de aplicaciones es mejorar la calidad de vida de los usuarios que las utilizan, finalmente 5) *Debe contener solo los elementos necesarios*, es decir, debe ser una interfaz limpia que emplea elementos solo para su funcionamiento

Por lo tanto, la app de Diabetes II figura como un cambio, esta nueva aplicación tiene un carácter cambiante y dinámico, que es funcional a los propósitos y objetivos para un mayor control sobre los cuidados y prevenciones, que deben tener en cuenta los usuarios finales, así como una mejor administración de sus hábitos tanto alimenticios como físicos, respaldados por un conjunto de asociaciones médicas sobre la Diabetes.

### **Solución propuesta**

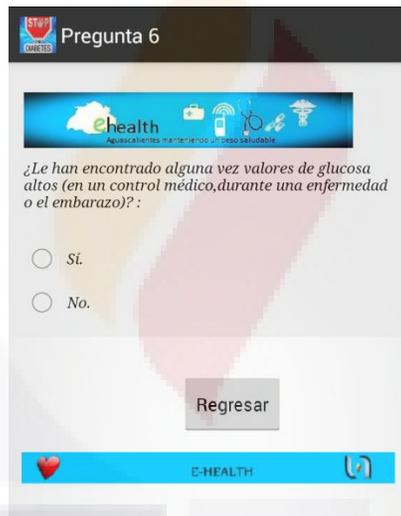
La aplicación “Diabetes II” (disponible en: <https://play.google.com/store/apps/details?id=test.diabetes>) a través de Google Play, se constituye por dos módulos de gran relevancia, la aplicación del test o cuestionario FINDRISK y el cálculo del Índice de Masa Corporal (IMC), los cuales fueron determinados en el análisis teórico de esta investigación.

El test FINDRISK de origen finlandés consta de ocho preguntas con puntuaciones predeterminadas, cuyo resultado predice la probabilidad personal de desarrollar diabetes a lo largo de los siguientes diez años, además el cuestionario nos permite conocer los hábitos alimenticios y físicos, así como antecedentes familiares y otros factores de riesgo. Un aspecto fundamental considerado para la elección de este cuestionario es, que ha demostrado ser eficaz para detectar el riesgo de diabetes en la población de distintos países europeos y está basado en la recaudación de información clínica y demográfica, que permite el cribado, como el auto-cribado no invasivo [4]. Dicho cuestionario ha sido traducido, adaptado y validado en distintas asociaciones de la diabetes y en numerosas campañas públicas de detección de la diabetes.

Existen otros cuestionarios para la población americana, como las basadas en los datos del San Antonio Heart Study o en el Atherosclerosis Risk in Communities (ARIC) study [5, 6]. Sin embargo, todas ellas adolecen de sencillez pues incorporan algún método invasivo como es la determinación de glucemia, lo que dificulta su aplicación fuera de la clínica por personal no sanitario. Además, el cuestionario Finandés cuesta solo unos tres de minutos de completar.

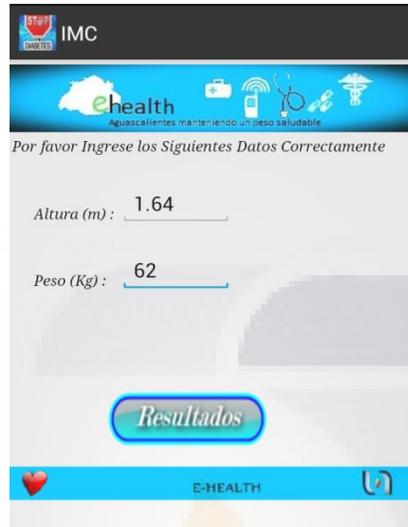
Por otro lado el cálculo del IMC, este cálculo está basado en la Norma Oficial Mexicana Para el tratamiento integral del Sobrepeso y de la Obesidad NOM-008-SSA3-2010 [7]. Cabe destacar que esta norma fue elaborada por un conjunto de dependencias e instituciones del Sector Salud Mexicano.

El resultado adquirido a través de la valoración de estos dos parámetros, es el diagnóstico que se le proporciona al usuario final, quien posteriormente incurre en la necesidad de cuidar su alimentación y realizar actividad física, como elementos claves para mejorar su salud y prevenir o retrasar la aparición de diabetes tipo 2. Cabe mencionar que el test FINDRISK no puede reemplazar un diagnóstico facultativo. Por este motivo, debe consultar con su médico el resultado obtenido.



**Fig. 1.** Pantalla de la interfaz de usuario.

Finalmente, al efectuar el cuestionario dentro de una aplicación en Android, se realizó con el fin de atender las características expuestas en el desarrollo de aplicaciones de e-health y así cumplir con un proceso automatizado, puesto que la evolución de tecnologías en móviles es un factor de vital importancia, por lo que la interfaz de usuario contiene imágenes referentes al tema de la salud, se muestra la pregunta y las posibles respuestas, logrando así, mayor integridad en las interfaces y mejoramiento en el manejo y procesamiento de datos, en la Figura 1 y 2, se puede apreciar el prototipo de la interfaz de usuario.



**Fig. 2.** Pantalla del calculo del IMC

Las interfaces del cuestionario, son lo más claras y precisas, con solo elementos necesarios y simples, esto para que el usuario final complete satisfactoriamente el cuestionario, sin algún factor de distracción sobre el objetivo principal de esta aplicación, que en este caso es el diagnóstico preventivo. Ya completado el cuestionario se pide al usuario que ingrese su altura y su peso, para poder calcular su IMC. Finalmente en base a estos dos módulos, se muestra el resultado al usuario, en la Figura 3 se muestra el prototipo de la pantalla de resultados.



**Fig. 3.** Diagnóstico del riesgo de adquirir DM-II

El diagnóstico que se muestra en la Figura 3, se conforma del total de puntos existentes, obtenidos de la valoración de los parámetros anteriormente expuestos, posteriormente de acuerdo al calculo del IMC se muestra el grupo donde se encuentra, acorde con el peso actual, esto conforme a la Norma Oficial Mexicana para el Tratamiento Integral del Sobrepeso y de la Obesidad NOM-008-SSA3-2010 [7], seguido del valor numérico que arroja el cálculo del IMC. A continuación, se presenta el nivel de riesgo en que se encuentra el usuario de adquirir DM-II, con las leyendas “Riesgo bajo”, “Riesgo ligeramente aumentado de adquirir DMII”, “Riesgo Moderado de adquirir DMII”, “Riesgo Alto de adquirir DMII” y “Riesgo Muy Alto de adquirir DMII” según sea el caso. También, se incluye en la siguiente línea la leyenda “*El test FINDRISK no reemplaza un diagnóstico médico. Por este motivo, debe consultar a su médico el resultado obtenido.*”

Además, se exponen un par de consejos para una vida más saludable, los cuales ayudan a prevenir o retrasar el padecimiento de la Diabetes, cabe mencionar, están sustentados y validados por un conjunto de Asociaciones de la Salud [1]. En la Figura 4 se muestra el prototipo de la pantalla de consejos.



**Fig. 4.** Interfaz de consejos

En la Figura 4, se describen los tipos de consejos que puede consultar el usuario, los cuales son: 1) *Factores de Riesgo*, es información que el usuario debe considerar para evitar contraer Diabetes tipo 2; 2) *Físicos*, muestra una imagen con diferentes deportes que el usuario puede realizar, junto con la leyenda “*Hacer ejercicio 30 minutos al día puede contribuir a la perdida de peso, lo cual a su vez puede reducir la presion sanguinea*”, y finalmente 3) *Alimenticios*, una imagen que nos indica los tipos de

alimentos que se pueden consumir de manera diaria, tres veces por semana o bien dos veces al mes.

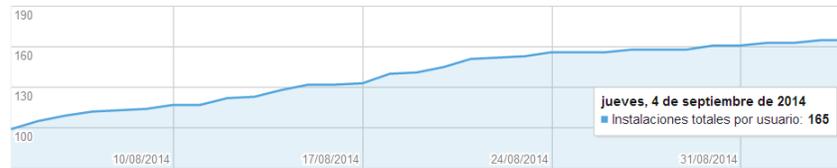
### Resultados

Los resultados y gráficas que se presentan en esta sección, fueron obtenidos a través de la Consola para Desarrolladores de Google, la cual permite dar un seguimiento adecuado de la aceptación y distribución de las aplicaciones publicadas, permite conocer las instalaciones actuales, valoración media, comentarios de los usuarios, etc. Como se mencionó anteriormente, la app DM-II se distribuye gratuitamente en todos los países donde Google Play tiene presencia, como se puede apreciar en la Figura 5, lo cual ha permitido que esta aplicación sea probada principalmente por usuarios de México, España, India, Brasil, etc. Cabe mencionar estos resultados se han logrado en un lapso de poco más de un mes de haber sido publicada.



Fig. 5. Países donde se ha descargado la app Diabetes II

Se han obtenido distintos comentarios de aceptación por parte de los usuarios, que van desde “muy bueno”, “Excelente, muy buena aplicación me sirvió mucho para mis tíos y abuelos”, hasta “Me parece una buena aplicación se las recomiendo a todos...te dice como mantener tu salud al 100% y para no correr riesgo de diabetes”. La aplicación Diabetes II fue publicada el 22 de Julio del año en curso. A la fecha la aplicación ha sido descargada un total de 165 ocasiones y cuenta con una valoración de 4.69 estrellas de un máximo de 5, como se puede observar en la Figura 6.



**Fig. 6.** Descargas y valoraciones

Información obtenida al 05 de Septiembre del 2014, por el ritmo de crecimiento que se tiene registrado se espera que el número de descargas aumente, ya que no se han reportado errores de visualización o funcionamiento por la comunidad de usuarios.

### Conclusiones y trabajo futuro

En el presente trabajo de investigación, se implementaron las características básicas que pudieran considerarse para el desarrollo de aplicaciones de e-health, en este caso, se desarrolló una aplicación para concientizar a los usuarios sobre el diagnóstico preventivo y los cuidados necesarios para el tratamiento la Diabetes Mellitus tipo 2. Como se mencionó anteriormente, en ningún momento el resultado, consejos, diagnósticos e información que se presentan al usuario podrá sustituir la opinión de un especialista, al contrario se busca complementar los tratamientos tradicionales.

Los resultados obtenidos hasta el momento para la aplicación Diabetes II (ver en <https://play.google.com/store/apps/details?id=test.diabetes>), hacen que esta aplicación se encuentre catalogada en el rango de 100 a 500 descargas dentro de Google Play.

Se espera que en los próximos meses el número de descargas aumente, y con ello llegue esta información a más usuarios para que puedan fomentar la cultura de la concientización y de esta forma se pueda contribuir a mejorar su calidad de vida.

### Referencias

- [1]. International Diabetes Federation [IDF]. (2013b) ¿Qué es la diabetes? Recuperado el 18 de septiembre de 2013, de: <http://www.idf.org/worlddiabetesday/toolkit/es/gp/que-es-la-diabetes>
- [2]. Manual de Procedimientos Estandarizados para la Vigilancia Epidemiológica de la Diabetes Mellitus Tipo 2. Secretaria de Salud, México. Recuperado de: [http://www.epidemiologia.salud.gob.mx/doctos/infoepid/vig\\_epid\\_manuales/10\\_2012\\_Manual\\_DM2\\_vFinal\\_31oct12.pdf](http://www.epidemiologia.salud.gob.mx/doctos/infoepid/vig_epid_manuales/10_2012_Manual_DM2_vFinal_31oct12.pdf)
- [3]. Estadísticas a propósito del día mundial de la diabetes. Instituto Nacional de Estadística y Geografía. Recuperado de: <http://www.inegi.org.mx/inegi/contenidos/espanol/prensa/contenidos/estadisticas/2013/diabetes0.pdf>
- [4]. Lindström J, Tuomilehto J. The Diabetes Risk Score: A practical tool to predict Type 2 diabetes risk. Diabetes Care 2003; 26: 725-731.

- [5]. Stern MP, Williams K, Haffner SM. Identification of persons at high risk for type 2 diabetes mellitus: do we need the oral glucose tolerance test? *Ann Intern Med* 2002;136:575-81
- [6]. Schmidt MI, Duncan BB, Bang H, Pankow JS, Ballantyne CM, Golden SH, Folsom AR, Chambless LE; The Atherosclerosis Risk in Communities Investigators Identifying individuals at high risk for diabetes: The Atherosclerosis Risk in Communities study. *Diabetes Care* 2005;28:2013-8)
- [7]. NOM-008-SSA3-2010, Norma Oficial Mexicana Para el tratamiento integral del Sobrepeso y de la Obesidad. (Fuente: Diario Oficial de la federación de fecha 04 de Agosto de 2010).



## Desarrollo de Aplicaciones Móviles para e-health: Aplicación Test de Obesidad

Omar Mariscal Pedraza<sup>1</sup>, María Elena Solorio Mejía<sup>2</sup>, Carlos Alberto Veloz Vidal<sup>3</sup> y  
Fracisco Javier Alvarez Rodríguez<sup>3</sup>

<sup>1</sup> Instituto Tecnológico de Morelia, Dpto. Ingeniería Electrónica, Av. Tecnológico  
#1500, Col. Lomas de Santiaguito, C.P. 58120, Morelia, México.  
omarmape@gmail.com

– <sup>2</sup> Instituto Tecnológico de Tepic, Dpto. Sistemas y Computación, Av. Tecnológico  
#2595, Col. Lagos del Country, C.P. 63175, Tepic, México.  
– elenasolorio9204@gmail.com

<sup>3</sup> Universidad Autónoma de Aguascalientes, Centro de Ciencias Básicas, Dpto. Sis-  
temas de Información Av. Universidad #940, Cd. Universitaria, C.P. 20131, Aguasca-  
lientes, México.  
carlosa.veloz@gmail.com, fjalvar@correo.uaa.mx

**Resumen.** Las enfermedades crónico degenerativas como el sobrepeso y la obesidad han ido aumento en los últimos años a nivel mundial afectando a el grueso de la población mundial, por lo que una manera de hacer consiente y responsable a la población sobre sus hábitos alimentación y actividad física, es acercarla y ponerle a la mano información de diagnóstico, prevención y tratamiento de estos padecimientos. Es por ello que se desarrolló una aplicación para dispositivos móviles para el sistema operativo Android, la cual brinda un pre-diagnóstico y da un primer acercamiento a todo aquel que descargue la aplicación, conteste el cuestionario y brinde los datos requeridos. Durante el desarrollo de busco brindar la suficiente información objetiva con los parámetros utilizados normalmente por los médicos especialistas en el área.

**Palabras clave:** dispositivos móviles, e-health, obesidad – sobrepeso, IMC, GEB.

### 1 Introducción

Hoy en día la tecnología ha tomado un papel primordial en la vida de casi todos los seres humanos. La mayoría de los aparatos eléctricos y electrónicos que utilizamos a diario, han sido dotados de características tales que sus funciones son capaces de sobrepasar nuestras necesidades.

Un ejemplo de lo descrito anteriormente es el teléfono celular o mejor conocido como teléfono inteligente, concebido en sus inicios con la única finalidad de realizar llamadas telefónicas sin comprometer la movilidad del usuario. Actualmente, ha evolucionado a tal nivel que puede realizar tareas equiparables a las de una computadora de escritorio. Por esto mismo su uso ha pasado de ser un lujo a una necesidad; siendo cada

vez mayor el número de personas poseedoras de uno. Se calcula que a finales del 2014 habrá 7 mil millones de abonos a telefonía móvil, y puesto que la tasa de crecimiento alcanza niveles cada vez más bajos (actualmente 2,6% en todo el mundo), se cree que el mercado está llegando a un punto de saturación [1].

Un punto importante que ha aportado a la popularidad de los celulares es el hecho que el usuario puede personalizar las funciones del dispositivo mediante la instalación de apps (aplicaciones), mismas de las que puede prescindir con facilidad. Esto ha generado que los desarrolladores busquen crear aplicaciones para necesidades cada vez más específicas, mismas que antes se creía no debían ser cubiertas por las tareas de un celular. Una de estas necesidades es la salud.

Recientes innovaciones tecnológicas están cambiando el contexto del cuidado de la salud y la administración médica para enfermedades no transmisibles y proporciona a la comunidad global nuevas oportunidades para la prevención y control de las mismas [2]. Una de estas innovaciones es conocida como mHealth, la cual se refiere al uso de comunicaciones móviles y tecnología multimedia para apoyar en la prestación de servicios y sistemas de salud [3].

Lo anterior representa la principal motivación para la realización de este trabajo, cuyas secciones se distribuyen de la siguiente manera: la sección 2, presenta la problemática actual en las apps de e-health relacionadas con el tema de la obesidad y el sobrepeso, seguida de la solución propuesta y el prototipo generado a partir de las características mencionadas en la sección 3, los resultados parciales de la app desarrollada “Test de Obesidad” disponible en la plataforma Google Play se presentan en la sección 4. Finalmente las conclusiones, así como el camino a seguir en el corto plazo se presentan en la sección 5.

## 2 Problemática

La OMS (Organización Mundial de la Salud) define como obesidad cuando el IMC (índice de masa corporal, cociente entre la estatura y el peso de un individuo al cuadrado) es igual o superior a 30 kg/m<sup>2</sup>. También se considera signo de obesidad un perímetro abdominal en hombres mayor o igual a 102 cm y en mujeres mayor o igual a 88 cm. La obesidad forma parte del síndrome metabólico, y es un factor de riesgo conocido, es decir, es una indicación de la predisposición a varias enfermedades, particularmente enfermedades cardiovasculares, diabetes mellitus tipo 2, apnea del sueño y osteoartritis, así como para algunas formas de cáncer, padecimientos dermatológicos y gastrointestinales [4].

Los teléfonos móviles han sido utilizados con éxito en el campo de las enfermedades transmisibles y la salud materno/infantil para mejorar el acceso a los servicios médicos para capacitar a trabajadores de la salud, para asegurar el cumplimiento del tratamiento (en el seguimiento y vigilancia) y en el tratamiento de enfermedades crónicas. En el campo de las enfermedades no transmisibles hay buena evidencia sobre el tratamiento de enfermedades utilizando apps [2].

Existen miles de aplicaciones disponibles en la Google Play Store listas para ser descargadas por los usuarios, sin embargo menos del 1% del total de aplicaciones dis-

ponibles se encuentran dentro de la categoría de Medicina. Al realizar una búsqueda en Google Play con las palabras: “*control de peso*”, “*obesidad test*” y “*obesidad diagnostico*”, se observa en los resultados que son muy pocas las aplicaciones que son desarrolladas (o por lo menos respaldadas) por instituciones o especialistas, ya que la gran mayoría de aplicaciones disponibles son creadas por desarrolladores independientes.

Esta situación representa un problema importante debido a que los resultados, diagnósticos, sugerencias/consejos, dietas, y demás resultados generados por las apps son poco confiables. Esto a la larga podría verse reflejado directamente en los volúmenes de descarga de las apps disponibles, al ser percibidas como poco confiables por los usuarios desde la lectura de la descripción en los repositorios como Google Play Store. Otro aspecto que puede percibirse como un punto de mejora, es sin duda que la mayoría de las aplicaciones relacionadas con obesidad o sobrepeso, se enfocan en presentar al usuario una serie de consejos o dietas que con el tiempo le permitan llegar a su peso “*ideal*”, el cual en muchas apps es establecido por él mismo usuario pudiendo desencadenar un problema de salud adicional. Un número menor de apps se orientan al seguimiento del peso del usuario a través de la captura periódica de su peso, y otra información indicando si se está o no progresando en el logro de las metas establecidas. Al igual que en el caso anterior, dichas metas son precisadas por el mismo usuario o por valores proporcionados por la app, basados en tablas que no indican sus orígenes lo cual nuevamente representa un riesgo para la salud, generando resultados contraproducentes.

Con el fin de contribuir a reducir esta brecha, se busca desarrollar una aplicación que emplee un cuestionario de hábitos alimenticios, un módulo para el cálculo del IMC y lo relacione con el GEB (Gasto Energético Basal), de esta manera se tiene información sobre el estilo de vida y consumo de alimentos del usuario, información con la cual se puede generar un diagnóstico que promueva la concientización entre los usuarios que utilicen esta app y al mismo tiempo sirva como información adicional al médico especialista en el área.

### **3 Desarrollo de la aplicación Test de Obesidad**

Teniendo en cuenta la seriedad con la que deben ser tratados estas enfermedades, se buscó el desarrollo de una aplicación que exclusivamente orientara al usuario indicándole que tanto riesgo tiene de padecer sobrepeso u obesidad, además de recomendar acudir con un especialista para un diagnóstico y tratamiento adecuados. Para ello se seleccionó el cuestionario desarrollado por Universidad de Valencia que consta de 37 reactivos que evalúan 8 factores [5]. Se utilizaron los parámetros del IMC (Índice de Masa Corporal) y GEB (Gasto Energético Basal) comúnmente utilizados en el diagnóstico y tratamiento de dichos padecimientos. A continuación se describen las características del módulo que calcula el GEB y del módulo que implementa el cuestionario de hábitos alimenticios.

### 3.1 Gasto Energético Basal (GEB)

La ecuación de Harris-Benedict es una ecuación empírica para estimar el metabolismo basal de una persona en función de su peso corporal, estatura y edad, y es utilizado en conjunto con factores de actividad física para calcular la recomendación de consumo diario de calorías para un individuo [6 y 7]. La ecuación supone una composición corporal normal, con una relación media entre la masa muscular y la masa grasa, por lo que puede ser inexacta para las personas que son muy musculosas.

Las variables energéticas que se obtendrán de manera automática a través del mismo son: el gasto energético basal (GEB) y el gasto energético total (GET) que pueden calcularse por la fórmula de Harris- Benedict utilizando el peso actual del paciente.

- Hombres:  $GEB = 66 + 13.7 (\text{peso actual kg}) + 5 (\text{estatura, cm}) - 6.8 (\text{edad, años})$ .
- Mujeres:  $GEB = 655 + 9.6 (\text{peso actual kg}) + 1.7 (\text{estatura, cm}) - 4.7 (\text{edad, años})$ .

El GET se estima a partir de la suma del GEB, y de los factores de corrección (10% del efecto térmico de los alimentos y el nivel actividad física del paciente). El nivel de actividad física del paciente se determinará con base en la siguiente clasificación: sedentario, ligero, moderado y activo; la información para poder definir este parámetro de manera individualizada se deberá obtener del Formato de Antecedentes de Actividad Física y Valoración disponible en el Expediente Electrónico. Con base en el criterio asignado, se deberá consignar el valor correspondiente para el cálculo del GET de acuerdo a lo siguiente: Sedentario (factor=1) Ligera (factor=1.2) Moderado (factor=1.4) Activo (factor=1.6).

### 3.2 Cuestionario de hábitos alimentarios para pacientes con sobrepeso y obesidad

Diseñado para el tratamiento integral y personalizado para el control del peso, ya que es necesario que la recogida de información que se obtiene del paciente obeso sobre sus hábitos se haga de manera correcta.

El cuestionario utilizado consta de 36 preguntas que permiten conocer los hábitos alimentarios del paciente, dichas preguntas están clasificadas en tres grupos que son: 1) Hábitos alimenticios, 2) Actividad física y 3) Estado de ánimo. Un aspecto considerado para la elección de este cuestionario es, que a través de las preguntas realizadas se obtiene información sobre el consumo de azúcar en la dieta, incluso sobre el tipo de alimentos que frecuentemente consumen los usuarios con obesidad. Además, el cuestionario empleado, permite identificar de manera más precisa el estado de salud de un usuario ya que considera factores como el consumo de azúcar, alimentación saludable, ejercicio físico, contenido calórico, bienestar psicológico, tipo de alimentos, conocimiento y control y por último consumo de alcohol [5]. Se optó por elegir este cuestionario, ya que es fácil de implementar dentro de una aplicación para dispositivos móviles, permite establecer una valoración inicial sobre el estado de salud del usuario, la cual complementa los tratamientos preventivos tradicionales para el cuidado de la obesidad y el sobrepeso.

Teniendo en cuenta la seriedad con la que deben ser tratados estas enfermedades, se buscó el desarrollo de una aplicación que exclusivamente orientara al usuario indicándole que tanto riesgo tiene de padecer sobrepeso u obesidad, además de recomendar

acudir con un especialista para un diagnóstico y tratamiento adecuados. Para ello se seleccionó el cuestionario desarrollado por Universidad de Valencia que consta de 37 reactivos que evalúan 8 factores [9], se utilizaron los parámetros del IMC (Índice de Masa Corporal) y GBE (Gasto Energético Basal) comúnmente utilizados en el diagnóstico y tratamiento de dichos padecimientos.

Primeramente se recolecto la información y documentación necesaria para la sustentabilidad teórica de la aplicación, se realizaron los primeros bocetos de lo que haría y como luciría la aplicación, validando estas hipótesis con un médico especialista en Medicina Integrada, se realizaron los cambios necesarios y se incluyeron los parámetros recomendados por el mismo. Posteriormente se desarrollaron las pantallas con las que interactúa el usuario como se aprecia en la Figura 1 para finalmente y una vez terminada la aplicación, esta fue puesta en la plataforma de Google Play poniendo al alcance de los usuarios del sistema operativo Android de manera gratuita.



**Fig. 1.** Bocetos de la aplicación

Se obtuvo una aplicación muy parecida a lo esperado desde un inicio del desarrollo y diseño de la misma, con las limitantes que la plataforma posee, como se ve en la figura anterior, la interfaz es sencilla e intuitiva de utilizar para el usuario.

#### 4 Resultados

Las descargas de la aplicación han ido en aumento desde su publicación en la plataforma Google Play, como se muestra en la Figura 2 la aplicación Test de Obesidad ha sido descargado un total de 61 ocasiones.



**Fig 2.** Descargas de la app Test de Obesidad.

La app se distribuye de manera gratuita en todos los países donde Google Play tiene presencia, de los datos obtenidos a través de la consola para desarrolladores de Google muestran que esta app ha sido descargada principalmente en países de habla hispana como México, España, Argentina y sorpresivamente Brasil, siendo que la aplicación no cuenta con el soporte para el idioma portugués; así también es una sorpresa que aparezcan en la lista de descargas países como las Islas Caimán. Esta información se presenta en la gráfica de la Figura 3.



**Fig 3.** Distribución global de las descargas de la aplicación.

Desafortunadamente, los resultados no han sido los mejores ni los más sorprendentes si se comparan con otras aplicaciones comerciales como lo son los videojuegos; esto se puede atribuir a que la gran mayoría de los usuarios de teléfonos inteligentes no están acostumbrados ni se imaginan las herramientas a las que pueden acceder desde su equipo móvil, además, que el número de apps disponibles en la categoría de Medicina es menor al 1% del total de apps disponibles en Google Play.

### 5 Conclusiones y trabajo futuro

En la actualidad la inclusión, la apertura y el uso de las tecnologías de la información en el diagnóstico y acercamiento médico son una pieza clave y requieren de mayor desarrollo tecnológico para optimizar y romper con las barreras que se tienen. El uso de los dispositivos móviles como plataforma para el uso de aplicaciones de e-health, constituye una valiosa herramienta para reforzar y/o complementar los tratamientos tradicionales de diversos padecimientos, ya que la información que este tipo de apps muestran al usuario no pretende en ningún momento reemplazar la opinión del especialista, por el contrario, se busca que faciliten el tratamiento y prevención de diversos padecimientos. Como se mostró en el caso de estudio, es posible desarrollar una aplicación que brinde información de primera mano al usuario sobre su estado de salud, que le

permita dar un seguimiento y tratamiento adecuado sobre temas como el sobrepeso y la obesidad.

Sin duda la aplicación requiere contar con mayor desarrollo y mejora en su funcionamiento, a fin de proporcionar información veraz, objetiva y de la calidad que la proporcionaría un médico. Estas mejoras serán tomadas en cuenta de acuerdo a las opiniones de los usuarios de la plataforma Google Play.

## Referencias

- [1]. Unión Internacional de Telecomunicaciones. (2014). Recuperado de: [http://www.itu.int/net/pressoffice/press\\_releases/2014/23-es.aspx#.U-4TDvI5PD-](http://www.itu.int/net/pressoffice/press_releases/2014/23-es.aspx#.U-4TDvI5PD-)
- [2]. Organización Mundial de la Salud y Unión Internacional de Telecomunicaciones. (2012). mHealth for NCDx (WHO-ITU Joint Work-plan). Recuperado de: [http://www.who.int/nmh/events/2012/mhealth\\_background.pdf](http://www.who.int/nmh/events/2012/mhealth_background.pdf)
- [3]. Jonatha D. Payne. (2013). The State of Standards and Interoperability for mHealth. Recuperado de: [http://mhealthalliance.org/images/content/state\\_of\\_standards\\_report\\_2013.pdf](http://mhealthalliance.org/images/content/state_of_standards_report_2013.pdf)
- [4]. Organización Mundial de la Salud. (2014). Obesidad y sobrepeso. Recuperado de: <http://www.who.int/mediacentre/factsheets/fs311/es/>
- [5]. Castro Rodríguez P, Bellido Guerrero D, Pertega Díaz S. Elaboración y validación de un nuevo cuestionario de hábitos alimentarios para pacientes con sobrepeso y obesidad. *Endocrinología y Nutrición* 2010; 57 (4): 130-139.
- [6]. Fuentes Chaparro, Laiza; Protocolo para Orientación Nutricional en la prevención y control de enfermedades crónicas: Sobrepeso, Riesgo Cardiovascular y Diabetes, 2da Ed, México, D.F., Secretaria de Salud.
- [7]. Weir, JB. New methods for calculating metabolic rate with special reference to protein metabolism. *J Physiology* 1949; 109; 1-9.

## APLICACIÓN ANDROID CON FINES DE SALUD, BAJO EL CASO DE ESTUDIO DEL MONITOREO DE OBESIDAD

Viviana Bustos Amador, Carlos Alberto Veloz Vidal, Francisco Álvarez Rodríguez

Universidad Autónoma de Aguascalientes  
Centro de Ciencias Básicas, Dpto. Sistemas de Información  
Av. Universidad #940, Cd. Universitaria, CP. 20131  
Aguascalientes, México.

{ic.viviana.b, carlosa.veloz}@gmail.com, fjalvar@correo.uaa.mx

**Resumen.** Conforme a los avances tecnológicos y la gran aceptación del uso de dispositivos móviles, ha surgido la necesidad de generar aplicaciones para la plata-forma Android. Las aplicaciones de e-health para el control de peso que están disponibles para descargar de manera gratuita en la Google Play Store no representan una solución real al problema de obesidad ya que sus resultados no tienen un estudio médico que los respalde. El presente trabajo propone el desarrollo de una aplicación en Android para el Monitoreo de Obesidad, donde los resultados y recomendaciones mostrados al usuario están avalados por especialistas en el área de la salud y conforme a lo establecido en NORMA Oficial Mexicana NOM-043-SSA2-2012, servicios básicos de salud, promoción y educación para la salud en materia alimentaria.

**Palabras clave:** dispositivos móviles, aplicación Android, e-health, obesidad.

### INTRODUCCIÓN

En años recientes el uso de teléfonos y dispositivos móviles se ha incrementado de manera tal que se han convertido en una alternativa al cómputo tradicional, haciendo más simple la movilidad, portabilidad y usabilidad dentro de la población en general; lo cual los convierte en una importante plataforma para crear aplicaciones relacionadas con la salud.

Los dispositivos móviles son una vía particularmente atractiva para el desarrollo de aplicaciones de la salud, debido a: (1) la adopción generalizada de dispositivos móviles con capacidades técnicas cada vez más potentes [1], (2) la tendencia de las personas a llevar estos dispositivos con ellos a todas partes, (3) el apego de las personas a sus teléfonos [2], y (4) las características de sensibilización de contexto habilitadas a través de sensores y la información personal vía telefónica.

Ya que se cuenta con este gran recurso, se busca aplicar sus beneficios para apoyar en actividades multidisciplinarias como lo es la salud, en este trabajo se presenta la creación de una aplicación que permite a los usuarios conocer en qué situación de salud se encuentran, con respecto a un análisis de su estilo de vida, en base a su actividad física, alimentación, estado de ánimo e índice de masa corporal (IMC).

El trabajo propuesto en este documento surge de un proyecto de nivel maestría, donde se busca que los alumnos adquieran experiencia en la realización de un proyecto y abordar todas las etapas que conforman el ciclo de vida para el desarrollo de un sistema, la idea es proponer y dejar un antecedente para que otros alumnos retomen y mejoren dicho proyecto, o incluso propongan nuevas ideas. Se pretende generar ideas innovadoras que tengan algún impacto dentro de la sociedad en múltiples áreas, considerar los problemas existentes y proponer herramientas de apoyo o posibles soluciones que sean accesibles para un mayor número de usuarios, mejorando así su calidad de vida.

El resto del artículo se organiza de la siguiente manera. Sección II presenta la problemática. Los antecedentes se presentan en la sección III. La solución propuesta se describe en la sección IV, mientras que las conclusiones y trabajo futuro se presentan en la sección V.

## **PROBLEMÁTICA**

Actualmente México ocupa el primer lugar mundial en obesidad en la población adulta (30%) por encima de Estados Unidos (28%) [3]. Con respecto a la población infantil, México ocupa el cuarto lugar de prevalencia mundial de obesidad, aproximadamente 28.1% en niños y 29% en niñas, superado solo por Grecia, Estados Unidos e Italia [4].

La obesidad tiene diferentes orígenes como lo son la predisposición genética, estilo de vida, falta de actividad física, el insuficiente consumo de frutas y verduras, la cultura, la educación, la globalización, el entorno político y social, por mencionar algunos de los principales factores. Básicamente el problema de obesidad y sobrepeso se resume en un desequilibrio energético entre las calorías que se consumen y las que se gastan. En todo el mundo se ha incrementado la ingesta de alimentos hipercalóricos (ricos en grasa, sal y azúcares pero pobres en vitaminas, minerales y micronutrientes) y se ha reducido la actividad física (como resultado de un estilo de vida más sedentaria, ya sea por el trabajo, cambio en los medios de transporte y la creciente urbanización) [5].

En el transcurso de la investigación sobre las aplicaciones Android existentes de e-health en la Google Play Store, resalta el hecho de que existan pocas aplicaciones relacionadas con la obesidad, otra característica es que estas aplicaciones no tienen el respaldo de instituciones o especialistas en el área, además de que el cálculo de IMC y la captura periódica del peso del usuario, no es algo concluyente para proporcionar un diagnóstico correcto o significativo al usuario.

## **ANTECEDENTES**

Existen miles de aplicaciones disponibles en la Google Play Store listas para ser descargadas por los usuarios, pero solo un porcentaje reducido se encuentran dentro de la categoría de Medicina, menos del 1% del total de aplicaciones disponibles. Se realizó una búsqueda con las palabras clave “control de peso” y se tomaron las 5 aplicacio-

nes mejor valoradas para determinar sus características y así poderla comparar con la solución propuesta. A continuación en la Tabla 1 se describen a detalle las aplicaciones analizadas.

Aplicaciones/ Características	Moni- torea Tu Peso[6]	Con- trol de Peso [7]	Perder peso, ad- elgazar, IMC [8]	Peso Asistent e [9]	Peso Dia- ry [10]	Moni- toreo de Obesidad [11]
Multiusuario	X	X	X			
Calculo IMC	X	X	X		X	X
Calculo BRM					X	
Calculo ingesta diaria de calorías	X					
Calculo % grasa cor- poral	X	X	X			
Peso perdido/ por perder	X					
Promedio día/semana	X					
Gráficas para se- guimiento	X	X	X	X	X	
Guardar/exportar resultados	X	X	X			X
Define objetivos (per- der o ganar peso)		X		X	X	
Captura Peso de mane- ra periódica			X	X	X	
Publicidad		X				
Test basado en hábitos alimenticios, estado de ánimo y actividad física						X
Información respalda- da por especialistas						X
Proporciona clasifi- cación de resultados						X
Proporciona ciertas recomendaciones de salud						X

**Tabla 2.** Comparativa de aplicaciones para control de peso.

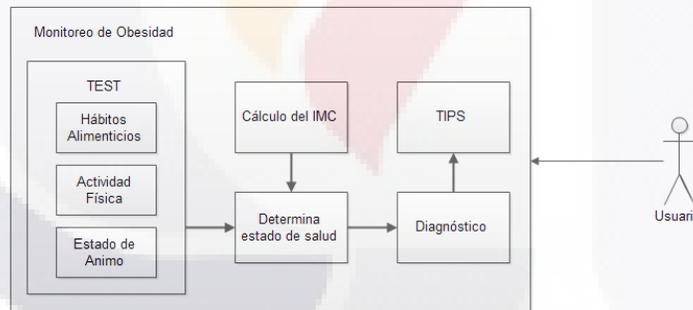
Uno de los aspectos importantes a mencionar es que las aplicaciones descri-tas son gratuitas; además resalta el hecho de que no hay un estudio médico detrás que avale los resultados que se muestran al usuario, al menos así se puede apreciar según la ficha técnica disponible en cada una de ellas; todas estas aplicaciones se enfocan en darle un

seguimiento al peso del usuario a través de invitarlo a que capture sus datos de manera periódica. Pero esto no es adecuado, ya que el usuario con el fin de ver un progreso puede ingresar datos falsos.

Todos los resultados mostrados en cada una de las aplicaciones analizadas se prestan a la interpretación que el usuario les dé; ya que no hay un seguimiento oportuno ni recomendaciones que lo orienten hacia un estilo de vida más saludable; por otra parte los resultados y recomendaciones que se presentan en la aplicación Monitoreo de Obesidad están avalados por la Norma Oficial Mexicana NOM-043-SSA2-2012 [12]; esta norma fue elaborada por distintas unidades administrativas e instituciones de renombre en el sector salud mexicano, donde se busca establecer los criterios para orientar a la población hacia una alimentación saludable de acuerdo a sus necesidades y capacidades.

### SOLUCIÓN PROPUESTA

Uno de los aspectos principales de las aplicaciones analizadas en la Tabla 1, es la carencia de un estudio científico o bien que profesionales en el área de la salud respalden los resultados que se muestran al usuario. Para solucionar esta problemática se desarrolla la aplicación Monitoreo de obesidad, que se compone de un módulo para la aplicación del test de hábitos alimenticios, otro para el cálculo del IMC, en base a esta información del usuario se puede proporcionar un diagnóstico de su estado de salud y en base a este, presentar una serie de tips que lo orienten hacia un estilo de vida más saludable. En la Figura 1, se muestre un diagrama de la aplicación a desarrollar.



**Fig. 1.** Diagrama de la aplicación Monitoreo de Obesidad

El test que se implementó en la aplicación consta de 36 preguntas, las cuales se pueden clasificar en tres grupos que son: 1) Hábitos alimenticios, 2) Actividad física y 3) Estado de ánimo. Además, este cuestionario permite identificar de manera más precisa el estado de salud de un usuario ya que considera factores como el consumo de azúcar, alimentación saludable, ejercicio físico, contenido calórico, bienestar psicológico, tipo de alimentos, conocimiento y control y por ultimo consumo de alcohol [13].

Al trasladar el cuestionario a una aplicación en Android, se implementó una interfaz limpia, libre de cualquier elemento que pudiera molestar al usuario como el uso de publicidad, dejando solo la pregunta y las opciones disponibles para cada una de ellas, junto con un botón de regresar en caso de que el usuario hubiera seleccionado por error alguna respuesta diferente a la que tenía pensado. En la Figura 2 se pueden apreciar algunas pantallas de las interfaces a utilizar.

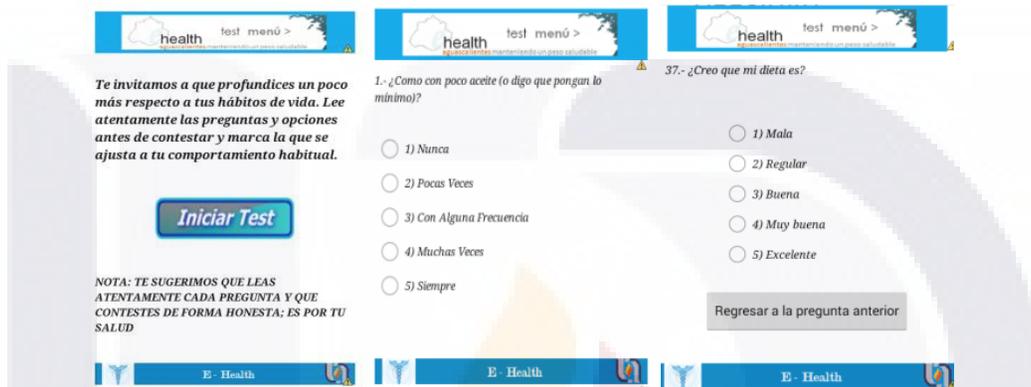


Fig. 2. Pantallas de las interfaces de Monitoreo de Obesidad

Una vez que ha concluido el test se pide al usuario que ingrese su altura y su peso, para poder calcular su IMC. Con estos dos módulos mencionados es posible calcular la cantidad de calorías ingeridas y el porcentaje de grasa corporal, pero lejos de presentar una serie de números que no tienen un significado que el usuario pueda asimilar, se muestra un diagnóstico real de su situación actual. En la Figura 3 se implementa la siguiente interfaz donde se muestran el diagnóstico al usuario.

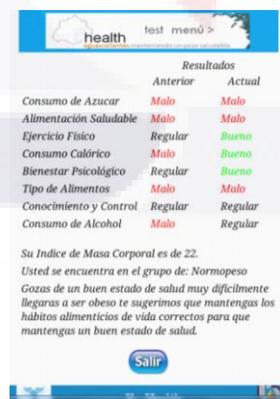


Fig. 3. Diagnóstico del estado de salud

El diagnóstico está integrado por diversas dimensiones, que le indican al usuario su resultado clasificando diversas características de sus hábitos alimenticios, de actividad física y estado de ánimo. Cada una de las preguntas contiene 5 opciones de respuesta, mismas que adquieren un valor definido, conforme al consumo, la frecuencia, etc. Al final del test, se genera una organización de preguntas conforme a la dimensión que corresponda, por ejemplo el consumo de azúcar, los tipos de alimentos, etc. El resultado se cuantifica y se genera un número entero que representa su diagnóstico por dimensión adquiriendo un valor ideal, mínimo o de riesgo. En base al resultado, se establecen los tips de acuerdo a cada rango de valores, los cuales pueden ser para motivar, cuando se adquiere un valor mínimo en ciertas dimensiones; para motivar y concientizar, cuando se presentan valores de riesgo como se puede apreciar en la Tabla 2.

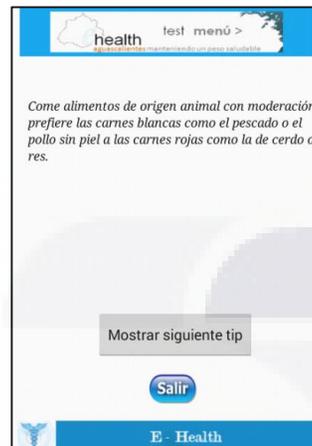
<i>Dimensión</i>	<i>Valor ideal</i>	<i>Valor mínimo</i>	<i>Valor en riesgo</i>
<i>Consumo de azúcar</i>	$\geq 19$	17-18	$< 17$
<i>Alimentación saludable</i>	$\geq 42$	37-41	$< 37$
<i>Ejercicio físico</i>	$\geq 13$	9-12	$< 9$
<i>Contenido calórico</i>	$\geq 21$	17-20	$< 17$
<i>Bienestar Psicológico</i>	15	12-14	$< 12$
<i>Tipo de alimentos</i>	$\geq 23$	19-22	$< 19$
<i>Conocimiento y control</i>	22	18-21	$< 18$
<i>Consumo de alcohol</i>	$\geq 10$	9	$< 9$

**Tabla 3.** Dimensiones del test realizado y rango de valores para los tips

Uno de los valores agregados más importantes de este caso de estudio es el módulo de tips, en el cual se establecen ciertas recomendaciones de salud proporcionados por la Norma Oficial Mexicana NOM-043-SSA2-2012, los cuales se pueden utilizar en diferentes dimensiones, bajo esta propuesta se muestra un ejemplo de la aplicación de tips en la Tabla 3 y en la Figura 4.

<i>Dimensión</i>	<i>Valor ideal</i>	<i>Valor mínimo</i>	<i>Valor en riesgo</i>
<i>Alimentación saludable</i>	En cada una de las comidas del día incluye al menos un alimento de cada uno de los tres grupos y de una comida a otra varía lo más posible los alimentos que se utilicen de cada grupo, así como la forma de prepararlos	-Incluye cereales integrales en cada comida, combinados con semillas de leguminosas.  -Come alimentos de origen animal con moderación, prefiere las carnes blancas como el pescado o el pollo sin piel a las carnes rojas como la de cerdo o res.  -Toma en abundancia agua simple potable.	-Incluye cereales integrales en cada comida, combinados con semillas de leguminosas.  -Come alimentos de origen animal con moderación, prefiere las carnes blancas como el pescado o el pollo sin piel a las carnes rojas como la de cerdo o res.  -Toma en abundancia agua simple potable.

**Tabla 4.** Ejemplo de los tips para la dimensión de Alimentación Saludable



**Fig. 4.** Pantalla donde se muestran los tips

En el transcurso de este proyecto se ha llevado a la práctica gran parte de las actividades necesarias y establecidas para el diseño y desarrollo de sistemas, cabe mencionar que este ha sido un importante avance ya que la manera en la que se aplican estos pasos al desarrollo de aplicaciones Android resulta por demás interesante, es importante que se busquen nuevas formas de aprovechar la tecnología en este caso de estudio es claro que se busca realizar una aportación en el área de salud.

Cada día es más grande el grupo de personas dispuestas a utilizar la tecnología, que mejor contribución que generar una herramienta para que conozcan el estado de salud en que se encuentran.

## CONCLUSIONES Y TRABAJO FUTURO

Aunque existen miles de aplicaciones Android que se distribuyen a través de la Google Play Store, un porcentaje reducido se encuentra disponible para el área de la salud, lo que permite que este campo aún se pueda explotar más en beneficio de los usuarios que las utilizan. Es importante que todas las aplicaciones que pretendan atacar un problema relacionado con las ciencias de la salud tengan un respaldo científico, ya que de lo contrario, se prestan a la interpretación y los resultados que se presenten al usuario no serán adecuados a su condición actual.

El presente caso de estudio proporciona un ejemplo de que es posible la contribución en el área de salud mediante el uso de los dispositivos móviles, los módulos que se emplearon para su desarrollo cuentan con estudios científicos que avalan la veracidad de los resultados que se presentan al usuario ya que los tips para mejorar su calidad de vida están basados conforme lo marca la norma oficial mexicana para una mejor alimentación de la sociedad mexicana. Dado que la obesidad es un problema que no solo

aqueja a México sino que tiene un impacto a nivel global, y los factores que la originan son prácticamente los mismos permite que la aplicación se pueda distribuir de manera abierta al resto de los países donde Google Play Store tiene presencia. Como trabajo futuro se planea desarrollar una plataforma de e-health que funcione como un canal de comunicación para usuarios y especialistas, lo cual reduzca la brecha que impide que este tipo de padecimiento sea atendido de manera oportuna para la sociedad Aguascalentense.

## REFERENCIAS

1. Pew Internet & American Life Project. Internet, broadband, and cell phone statistics; 2010. <http://www.pewinternet.org/Reports/2010/Internet-broadband-and-cell-phone-statistics.aspx?r=1>
2. Ventä L, Isomursu M, Ahtinen A, Ramiah S. My Phone is a part of my soul – how people bond with their mobile phones. In: Proc UbiComm '08. IEEE Computer Society; 2008. p. 311–7.
3. Franco S. Obesity and the Economics of Prevention: Fit not Fat. Organization for the Economic Cooperation and Development (OECD publishing); 2010.
4. Franco S. Obesity Update 2012. Organization for the Economic Cooperation and Development (OECD publishing); 2012.
5. OMS (2006) Organización Mundial de la Salud. Obesidad y sobrepeso. Nota descriptiva no. 311 [en línea]. Mayo 2012. Disponible en <http://www.who.int/mediacentre/factsheets/fs311/es/index.html>
6. Hussain Al-Bustan. Monitorea Tu Peso [En línea] Disponible en <https://play.google.com/store/apps/details?id=monitoryourweight.bustan.net>
7. angelfmarcos. Control de Peso [En línea] Disponible en <https://play.google.com/store/apps/details?id=org.angel.weightcontrol>
8. cryofy.com Perder peso, adelgazar, IMC [En línea] Disponible en <https://play.google.com/store/apps/details?id=com.cryofy.android.weightmeterfree>
9. Kevin Tung. Peso Asistente [En línea] Disponible en <https://play.google.com/store/apps/details?id=com.ikdong.weight>
10. DSD. Peso Diary [En línea] Disponible en <https://play.google.com/store/apps/details?id=weight.manager>
11. Monitoreo de Obesidad. [En línea] Disponible en <https://play.google.com/store/apps/details?id=com.mipaquete.prototipo1&hl=es>
12. NOM-043-SSA2-2012, Servicios básicos de salud. Promoción y educación de salud en materia alimentaria. Criterios para brindar orientación. (FUENTE: Diario Oficial de la federación de fecha 28 de Mayo de 2012).
13. Castro Rodríguez P, Bellido Guerrero D, Pertega Díaz S. Elaboración y validación de un nuevo cuestionario de hábitos alimentarios para pacientes con sobrepeso y obesidad. Endocrinol Nutr 2010; 57 (4): 130-139.

Seleccione (de acuerdo a su percepción personal) el grado de cumplimiento para cada una de las siguientes 10 afirmaciones.

**1. El proceso para el diseño de aplicaciones nativas en Android basado en mejores prácticas y metodologías ágiles, cuenta con un adecuado soporte científico/teórico.**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**2. El marco teórico utilizado es relevante para el diseño de aplicaciones nativas en Android**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**3. Cada una de la etapas del proceso son coherentes**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**4. El proceso es adecuado para lograr el propósito de su utilización**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**5. Si utilizara el proceso, podría lograr diseñar una aplicación nativa en Android de manera más eficiente**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**6. El proceso es complejo, y difícil de utilizar**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**7. Si utilizara el proceso, simplificaría el trabajo de diseñar aplicaciones nativas en Android**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**8. Considera que aprender a usar el proceso, es sencillo**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**9. Si usara el proceso, le resultaría fácil de adaptar a sus necesidades**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

**10. Considera que proceso sería difícil de usar en la práctica**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo

**11. Consideraría utilizar el proceso presentado en futuros proyectos para desarrollar aplicaciones nativas en Android**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo

**12. Preferiría utilizar otras estrategias para diseñar aplicaciones Android, antes que utilizar el proceso presentado**

Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo

