



Instituto Tecnológico Superior de Martínez de la Torre

Impacto de las aplicaciones móviles
como herramienta de promoción y
publicidad dentro del ITSMT

TESIS

PARA OBTENER EL GRADO:

Ingeniería en Sistemas
Computacionales

PRESENTA:

Aldo Isaías Albor León

ASESOR:

MRYSI. Ángel Salas Martínez

Martínez de la Torre, Ver.

DICIEMBRE, 2019



**TECNOLÓGICO
DE MARTÍNEZ**

Nombre del Documento:
Carta de Autorización de Impresión

No. Pág.
1/1

Nº. Oficio DET/ITSMT/ISC/080/2018

ASUNTO: Autorización de Impresión TESIS.

Martínez de la Torre, Ver., a 26 de Septiembre de 2018

C. ALDO ISAÍAS ALBOR LEÓN
NO. DE CONTROL: 130I0087
EGRESADO(A) DE LA CARRERA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
P R E S E N T E

Por medio de la presente hago constar que ha cumplido satisfactoriamente con lo estipulado por el Lineamiento para la Titulación Integral.

Por tal motivo **Se Autoriza** la impresión de la Tesis titulada:

"Impacto de las aplicaciones móviles como herramientas de promoción y publicidad dentro del ITSMT."

Dándose un plazo máximo de 30 días naturales a partir de la fecha de la expedición de la presente para realizar la solicitud del Acto de Recepción para la obtención del Título Profesional.

A T E N T A M E N T E

MRSI. ÁNGEL SALAS MARTÍNEZ

Nombre y Firma
Presidente de Academia

MCA. YANAHUI BARRERA CASTELLANOS

Nombre y Firma
Jefe(a) de Carrera de Ingeniería

C.c.p. División de Estudios Profesionales.
C.c.p. Archivo



Asesor

MRYSI. Ángel Salas Martínez

Sinodales

ISC. Hugo Lucas Alvarado

MII. Karen Pamela Alvarez Villagómez

Índice general

Agradecimientos	4
Resumen	5
1 Introducción	7
1.1 Planteamiento del problema	8
1.2 Hipótesis	10
1.3 Objetivos	10
1.4 Justificación	11
1.5 Alcances y limitaciones	14
1.6 Organización del documento	14
2 Marco Teórico	17
2.1 Aplicaciones móviles	18
2.2 Tipos de aplicaciones móviles	20
2.3 Fases del diseño de aplicaciones móviles	23
2.4 Front-end y Back-end	30
2.5 Lenguajes de programación para dispositivos móviles	31
2.6 Sistemas operativos para dispositivos móviles	35
2.6.1 Sistemas operativos móviles principales	36
2.6.2 Android	37
2.6.3 IOs	40
2.7 Entorno de desarrollo para aplicaciones móviles Android	42
2.7.1 Android Studio	43
2.7.2 Eclipse	43
2.8 Bases de datos para dispositivos móviles	44
2.8.1 Ejemplos de bases de datos para dispositivos móviles	45

3 Estado del arte	49
3.0.1 Instituto Cumbres Mérida	49
3.0.2 MiTec Móvil	51
3.0.3 Guía UNAM	53
4 Marco metodológico	55
4.1 Enfoque de la investigación	55
4.2 Tipo de investigación	56
4.2.1 Según el propósito u objetivo de la investigación	57
4.2.2 Nivel de profundización del conocimiento	57
4.2.3 Según los tipos de datos a trabajar	58
4.2.4 Según el grado de manipulación de las variables	58
4.3 Alcance de la investigación	59
4.4 Diseño de la investigación	61
4.5 Cronograma de actividades	61
4.6 Unidades de análisis y técnicas de recolección de datos.	61
4.6.1 Unidades de análisis	62
4.6.2 Técnicas de recolección de datos	63
5 Desarrollo e Implementación de la app	65
5.1 Metodología de desarrollo de software a utilizar	65
5.1.1 Metodología de desarrollo de software general de Pressman	66
5.1.2 Propuesta de metodología de desarrollo de aplicaciones móviles a utilizar	67
5.2 Recolección y análisis de requerimientos	68
5.2.1 Etapa 1. Recolección y análisis de requerimientos.	69
5.2.2 Etapa 2. Diseño rápido	71
5.3 Plan y guía de trabajo	73
5.3.1 Etapa 3. Planeación	73
5.4 Diseño de la base de datos	79
5.4.1 Etapa 4. Modelado	80
5.5 Diseño de las pantallas de la app	82
5.6 Codificación front-end y back-end	85
5.6.1 Etapa 5. Construcción y desarrollo	86
5.6.2 Etapa 6. Pruebas y evaluación	100

5.7	Implementación de la app desarrollada	101
5.7.1	Etapa 7. Liberación	101
6	Experimentación y resultados	105
6.1	Actividad de recolección y análisis de datos	105
6.1.1	Enunciado 1	105
6.1.2	Enunciado 2	106
6.1.3	Enunciado 3	107
6.1.4	Enunciado 4	108
6.1.5	Enunciado 5	109
6.1.6	Enunciado 6	110
6.1.7	Enunciado 7	110
6.1.8	Enunciado 8	111
6.1.9	Enunciado 9	112
6.1.10	Enunciado 10	113
6.2	Actividad de observación e interpretación de resultados	114
6.3	Interpretación de resultados	115
7	Conclusiones	121
7.1	Conclusión	121
7.2	Trabajos Futuros	122
	Bibliografía	125

Índice de figuras

2.1	Comparación diseño básico entre aplicación nativa, basada en web y móvil. Fuente: [10]	23
2.2	Diagrama interpretativo del front-end y back-end. Fuente: [13]	31
2.3	Arquitectura de Android. Fuente: [9]	38
2.4	Arquitectura de IOs. Fuente: [26]	41
2.5	Arquitectura de base de datos móvil. Fuente [30]	45
4.1	Cronograma de actividades. Fuente: Elaboración propia.	61
5.1	Cronograma de actividades de desarrollo del proyecto. Fuente: Elaboración propia.	75
5.2	Diagrama de clases. Fuente: Elaboración propia.	76
5.3	Caso de uso de la app - Registro de preferencias. Fuente: Elaboración propia	77
5.4	Caso de uso del monitor - Administrador de usuarios. Fuente: Elaboración propia.	79
5.5	Diagrama por nodos de la app Difutec en forma de documentos. Fuente: Elaboración propia.	82
5.6	Diseño de pantalla del menú principal. Fuente: Elaboración propia.	84
5.7	Diseño de pantalla del modulo de consultas estadísticas. Fuente: Elaboración propia.	85
5.8	Resultado front-end de programación modulo consultas estadísticas en Android. Fuente: Elaboración propia.	88
5.9	Resultado del archivo index.php el cual contiene la maquetacion o codificación front-end del sistema monitor. Fuente: Elaboración propia.	95
6.1	Gráfica interpretativa enunciado 1. Fuente: Elaboración propia.	106
6.2	Gráfica interpretativa enunciado 2. Fuente: Elaboración propia.	107

6.3	Gráfica interpretativa enunciado 3. Fuente: Elaboración propia.	108
6.4	Gráfica interpretativa enunciado 4. Fuente: Elaboración propia.	109
6.5	Gráfica interpretativa enunciado 5. Fuente: Elaboración propia.	110
6.6	Gráfica interpretativa enunciado 6. Fuente: Elaboración propia.	111
6.7	Gráfica interpretativa enunciado 7. Fuente: Elaboración propia.	112
6.8	Gráfica interpretativa enunciado 8. Fuente: Elaboración propia.	113
6.9	Gráfica interpretativa enunciado 9. Fuente: Elaboración propia.	114
6.10	Gráfica interpretativa enunciado 10. Fuente: Elaboración propia.	115

Índice de cuadros

2.1	Tabla comparativa entre enfoque nativo, web e híbrido. Fuente:[10]	24
5.1	Tabla ilustrativa de los temas del capítulo en comparación con las etapas que se abordarán en cada uno. Fuente: Elaboración propia.	68
5.2	Separación de requisitos funcionales y no funcionales. Fuente: Elaboración propia.	71
5.3	Tabla de módulos propuestos, descripción y requisitos que pretenden cubrir. Fuente: Elaboración propia.	72
5.4	Tabla de calificación y pruebas de la aplicación en diferentes dispositivos. Fuente: Elaboración propia.	100
6.1	Checklist del investigador. Fuente: Elaboración propia.	116

Agradecimientos

A Dios por ser el guía de mi fé a traves de todo este largo camino.

A mis familiares y seres queridos por ser mi mas grande motivación, por creer en mí y brindarme siempre el apoyo y cariño que un padre le puede dar a un hijo o hermano.

A todas y cada una de las personas que logre conocer en mi estancia en ITSMT, a quienes para no errar omitiré mencionar, muchas gracias por su amistad.

¡ G R A C I A S !

Resumen

El desarrollo de aplicaciones móviles actualmente esta presentándose de manera exponencial, implementándose en muchos de los procesos que incluso anteriormente no se acababan de actualizar con un software de escritorio o web, en base al crecimiento y uso de las aplicaciones móviles que ayudan a la educación y muchos de los procesos que esta lleva consigo, es necesario medir su impacto como “herramienta” en estos procesos. Esto se debe a que sus implementaciones y desarrollos no cuentan con una metodología estandarizada que de como resultado un software de calidad, por ello un estudio posterior a su desarrollo en el ámbito para el que fue construida puede dar resultados sobre su nivel de aceptación en el proceso.

Múltiples metodologías que no son aceptadas aun, se implementan junto con el gran avance de las tecnologías nuevas o actualizadas que hoy en día están definiendo una nueva generación de desarrollo de software. El desarrollo de software de aplicación móvil, no es simplemente el pasar de varias versiones web y de escritorio a un dispositivo que cabe en el bolsillo. Desarrollar una aplicación que se libera al uso de cualquier persona o uso publico, conlleva la responsabilidad de estudiarse y analizar los resultados para evaluar los riesgos y las ventajas que le puede dar a un proceso.

Actualmente, hay muchas tecnologías que pueden ayudar a medir y estudiar el desarrollo de una aplicación. Google Inc, que se ha posicionado como principal proveedor de sistemas operativos para dispositivos móviles, implementa muchas tecnologías que a la par de otras tecnologías como las que desarrollan las compañías de redes sociales pueden dar como resultado una aplicación que no utiliza muchos recursos del dispositivo y ademas al utilizar tecnologías como Firebase pueden ayudar a monitorear el uso y obtener reportes estadísticos en tiempo real.

El presente trabajo de investigación tiene como objetivo documentar el estudio y resultados sobre una aplicación que se ha desarrollado como herramienta para mejorar un proceso de promoción y difusión de la oferta educativa del ITSMT, es importante

mencionar que esta aplicación se encuentra en su primer versión, aunque cuenta con algunas otras funciones de recopilación de datos y transformación de los mismos para generar información útil. Se va a realizar un estudio para medir su impacto en el proceso para el que se desarrolló la aplicación.

1 Introducción

La presente tesis es una investigación que tiene por objetivo evaluar el impacto que tienen las aplicaciones móviles funcionando como herramientas de promoción y publicidad en una institución universitaria, en este caso el Instituto Tecnológico Superior de Martínez de la Torre. Estas aplicaciones que se crearan para evaluar las hipótesis se conocen como aplicaciones móviles nativas puesto que estará disponible únicamente para el sistema operativo móvil Android. También hay otros tipos de aplicaciones que se abordarán durante el desarrollo para que se puedan analizar cómo es que realmente este tipo de aplicación impacta en este caso. Es necesario describir cuales son las fases por las que el diseño y desarrollo de una aplicación móvil tienen que pasar, de esta manera se comprende cómo es que el desarrollo y el contenido tendrán un impacto al objetivo planteado dentro de la investigación. Además de mencionar todos los lenguajes de programación para dispositivos móviles resaltando el porqué del lenguaje elegido sobre los demás y también el porqué del sistema operativo.

Al establecer lo anterior se puede comprender el entorno de desarrollo para las aplicaciones móviles sobre el sistema operativo elegido y el lenguaje de desarrollo. También es necesario mencionar la importancia de las bases de datos móviles y el tipo que se utilizará, sus ventajas y principales características para este tipo de aplicación y por qué utilizarse en estos casos.

Las fuentes bibliográficas provienen de investigaciones y artículos recientes del tema, principalmente artículos de los desarrolladores de estas tecnologías nuevas, utilizadas dentro del desarrollo de las aplicaciones por su grado de innovación y su impacto en el tema de los sistemas operativos móviles.

Este trabajo presenta los siguientes capítulos: En el capítulo I se presenta el planteamiento de la investigación o planteamiento del problema, las hipótesis, los objetivos, la justificación, los alcances y limitaciones de la misma y la organización del documento. En el capítulo II se presenta el marco teórico, todo aquel aspecto sobre las aplicaciones

móviles, los tipos de aplicaciones móviles, el tema descriptivo de front-end y back-end, los lenguajes de programación para dispositivos móviles y los sistemas operativos de los mismos, que entornos de desarrollo existen para aplicaciones móviles y las bases de datos que estos utilizan. En el capítulo III se presenta el estado del arte donde se incluirán todas aquellas investigaciones sobre aplicaciones similares que ya existen, sus objetivos, el impacto que han tenido y sus resultados. En el capítulo IV se presentará el proceso de desarrollo e implementación de la aplicación, una vez que se aborden los temas teóricos y se expliquen las ventajas y desventajas de estos, se pondrán en uso las tecnologías que sean de utilidad para el desarrollo de la aplicación de acuerdo a los requerimientos que se solicitan, el diseño que se ha establecido en cuanto a pantallas y la base de datos. En el capítulo V se ofrece la recolección, evaluación, discusión e interpretación de los resultados. En el capítulo VI se presenta la conclusión y líneas futuras del trabajo de investigación.

1.1. Planteamiento del problema

El Instituto Tecnológico Superior de Martínez de la Torre realiza de manera anual una jornada de promoción y difusión de la oferta académica con la que cuenta, además de promover los beneficios a los que se hacen acreedores los alumnos que forman parte de la familia ITSMT.

El programa de promoción y difusión se lleva a cabo en todas las comunidades cercanas a Martínez de la Torre, sin embargo, no a todos los institutos de educación media superior se pueden transportar todos los proyectos que se han desarrollado o que se encuentren en vías de desarrollo por estudiantes de las diferentes carreras que oferta el instituto tecnológico superior de Martínez de la Torre.

Los aspirantes a cursar una carrera universitaria en la mayoría de las ocasiones desconocen, que pueden hacer en cada una de las carreras o donde pueden laborar una vez que han concluido sus estudios.

El ITSMT solicita a los aspirantes que proporcionen algunos datos para enviar información relacionada con la carrera que les resulte de interés mediante correo electrónico. Con base en la información otorgada por los aspirantes, la institución puede identificar el interés del joven y proporcionar la información necesaria para que el aspirante considere el ITSMT como su primera opción, para continuar con su formación personal.

1.1 Planteamiento del problema

El desarrollo de una aplicación móvil para llevar a cabo la promoción y publicidad de la institución se podría contextualizar mediante escenarios como los que se presentan a continuación.

“Pedro es un alumno de bachillerato de sexto semestre, él está interesado en continuar sus estudios de nivel superior, sin embargo, él vive en una comunidad geográficamente alejada a Martínez de la Torre, motivo por el cual le resulta complicado asistir al instituto a solicitar información sobre la carrera que él desea cursar, el decide descargar la app para recibir información del instituto y elegir qué carrera desea cursar.”

“María termino su bachillerato hace algunos años, sin embargo, está decidida a retomar sus estudios de nivel superior, el ser una aspirante que ya no se encuentra dentro de una institución educativa, las visitas o muestras en instituciones educativas ya no serán una forma factible de que considere el ITSMT como una opción, hasta que se entera por algún conocido que puede descargar la app para consultar información sobre las carreras que puede cursar.”

“José es un alumno de educación media superior que está interesado en formar parte del ITSMT, pero no está seguro de que carrera elegir, estando presente en las visitas que el departamento de Difusión y Promoción del ITSMT hace a su escuela, se entera que puede realizar un test de orientación educativa sobre qué carrera elegir de acuerdo a sus aptitudes y actitudes.”

Para que escenarios como los anteriores sean posibles es necesario contar con una aplicación móvil que sirva para implementar la promoción, publicidad, difusión y al mismo tiempo sirva como herramienta institucional para identificar la preferencia de las carreras que los aspirantes presentan.

Las anteriores problemáticas podrían resumirse en que el Instituto Tecnológico Superior de Martínez de la Torre a pesar de tener una amplia distribución geográfica al momento de realizar visitas de promoción y difusión no puede abarcar todas las instituciones de educación media superior por lo que una herramienta de promoción y difusión podrían hacer llegar a más jóvenes la información necesaria sobre el ITSMT y su oferta académica.

Es importante recalcar también el problema que es elegir una carrera acorde a las aptitudes o actitudes que los jóvenes reflejan, por lo que el evaluar estas mediante un Test de Orientación Educativa puede solucionar el problema.

También una vez que los jóvenes han elegido al ITSMT como su primera opción de estudios es necesario que se mantengan informado sobre fechas, programas de estudio, eventos, proyectos, etc. Por lo que un apartado para notas informativas dentro de la herramienta podría reducir la publicación de notas y recordatorios en papel, además de que podrían llegar a más jóvenes cuyos dispositivos móviles tengan instalada la app. El ITSMT al no contar con una herramienta de análisis no pueden evaluar todos los datos que han recopilado de los jóvenes de las instituciones educativas para saber a futuro que carreras nuevas se pueden abrir.

1.2. Hipótesis

1. La promoción y difusión del ITSMT tendrá un mayor alcance geográfico debido a que la implementación de la aplicación móvil facilita el acceso a mas lugares de la región.
2. Los aspirantes estarán mejor preparados en su desempeño académico por que serán orientados para saber que carrera se ajusta a su perfil de acuerdo a sus Aptitudes y Actitudes.
3. Se conocerá la preferencia de los aspirantes en menor tiempo posible en función de las carreras que oferta el ITSMT.

1.3. Objetivos

Objetivo general

Desarrollar e implementar una aplicación móvil que sirva como herramienta para llevar a cabo la promoción y publicidad del Instituto Tecnológico Superior de Martínez de la Torre, además de generar estadísticas de preferencia sobre la oferta educativa.

Objetivos específicos:

- Desarrollar un primer prototipo funcional que permita cubrir las actividades y requisitos de promoción y publicidad del ITSMT, atendiendo al diseño de las metodologías de desarrollo de aplicaciones móviles utilizando herramientas de software libre.

- Estudiar herramientas y tecnologías para aplicaciones móviles.
- Generar un informe sobre las preferencias de la oferta educativa de la institución.
- Realizar un Test de orientación que permita identificar cual es la carrera afín a los jóvenes de acuerdo a sus aptitudes y actitudes.
- Diseñar una base de datos adecuada NoSQL para el almacenamiento de datos y procesamiento de la información.
- Diseñar una interfaz de usuario amigable que favorezca la interacción y navegación, además de ser intuitiva de acuerdo a las metodologías de desarrollo de aplicaciones móviles.
- Agilizar el proceso de registro de información de usuarios aspirantes para su análisis de preferencias educativas.
- Generar un monitor en plataforma web que permita a un usuario de tipo administrador el monitoreo de las opciones que la base de datos tiene.
- Generar reportes en forma de gráficos estadísticos para el usuario administrador y evaluar la preferencia de los usuarios aspirantes de acuerdo a distintos criterios.
- Generar reportes en forma de gráficos estadísticos para el usuario aspirante y que se permita visualizar la preferencia de los demás usuarios de la aplicación de acuerdo a distintos criterios.
- Desarrollar en tiempo y forma la aplicación móvil y sistema monitor.

1.4. Justificación

El Instituto Tecnológico Superior de Martínez de la Torre es una universidad de tipo publica de educación superior, se ubica en la ciudad de Martínez de la Torre en el estado de Veracruz, imparte licenciaturas de tipo ingeniería y es de control estatal.

Esta institución cuenta actualmente con 4 ubicaciones diferentes disponibles; En Martínez de la Torre se encuentran las instalaciones de Adolfo Ruiz Cortines, Ignacio de la Llave y Vicente Guerrero y en la localidad de Vega de Alatorre. En cada una de ellas se imparten distintas materias de acuerdo a la carrera. Las carreras disponibles actualmente son 5 las cuales son: Ingeniería en Sistemas Computacionales, Ingeniería

en Gestión Empresarial, Ingeniería Mecatrónica, Ingeniería Ambiental e Ingeniería en Industrias Alimentarias.

En esta institución educativa se busca implementar nuevas tecnologías para mejorar algunos de los procesos de promoción y difusión de la institución (visita de escuelas preparatorias para promocionar la institución, muestra de proyectos y actividades extraescolares, recopilación de datos para toma de decisiones, difusión de las principales actividades del ITSMT) así como de la promoción de la oferta académica para aquellas personas que aspiran a formar parte de la institución tomando alguna de las carreras ofertadas.

El proyecto a desarrollar para el ITSMT es una aplicación de promoción y difusión (Difutec) desarrollado en la plataforma de Android donde se incluyen los siguientes módulos; Módulo de registro de aspirante que recopila los datos de la persona que se registra, módulo de oferta académica donde se mostraran las carreras, actividades extraescolares y proyectos de la institución, módulo de test orientativo para el apoyo del aspirante en su elección, módulo de notas informativas para que los aspirantes a través de una visualización de Twitter puedan ver las notas más importantes que publica la institución, un módulo estadístico para que el aspirante pueda ver un reporte gráfico sobre las preferencias de la población y un módulo monitor en plataforma web para un administrador que pueda visualizar reportes de estadística y modificar opciones de la base de datos de la aplicación. Actualmente la institución realiza una promoción a solo algunas de las escuelas preparatorias de la región por lo que para un mayor impacto sin necesidad de visitar a todas las instituciones se puede utilizar esta aplicación móvil.

El proyecto se realiza también para implementar nuevas tecnologías que futuramente puedan seguirse mejorando y aportando más a la institución. Debido a los altos impactos que tienen las nuevas tecnologías de Android y las aplicaciones móviles se puede decir que es uno de los mejores caminos para promocionar a la institución, esto dará un beneficio institución – aspirante ya que la institución podrá hacer publicidad y difusión mediante la aplicación y todas aquellas personas de entre 17 y 24 años que desean tomar una carrera en esta institución podrán conocer la oferta académica. Las tecnologías móviles avanzan de manera exponencial por lo que su alcance en todo el mundo es mayor cada día, por ello una aplicación móvil puede ayudar a impactar más en la promoción de la institución sin necesidad de ir a todas las instituciones preparatorias de la región y mas allá de la región.

El motivo de implementar estas tecnologías es que actualmente se puede encontrar de 1 a 2 dispositivos móviles por persona entre los 17 y 24 años. De esas personas la mayoría utiliza el sistema Android y tendría acceso a la aplicación móvil para conocer los beneficios que brinda la institución.

La aplicación móvil de promoción y difusión del ITSMT (DIFUTECH) será una herramienta que va a facilitar el acceso a la información desde distintas partes, además de que se podrán realizar distintas tareas como; Test de orientación, consulta de oferta académica, consulta de proyectos, consulta de actividades extraescolares y visualización de notas informativas de la cuenta de Twitter de la institución, esto representa a la aplicación móvil Difutech como un instrumento de registro y consulta de información confiable y útil.

Algunas de las problemáticas que presenta la institución se listan a continuación:

- Al realizar visitas a escuelas preparatorias para promoción, ya que no se pueden visitar todas las escuelas de la región.
- Conflictos para hacer llegar información importante a jóvenes estudiantes pertenecientes a escuelas que no fueron visitadas.
- Dar a conocer los proyectos y actividades extraescolares para conocimiento de los jóvenes.
- Recopilar información sobre la preferencia de los aspirantes sobre las carreras que oferta el tecnológico.
- Análisis de información para la toma de decisiones sobre apertura de futuras carreras.
- Que los aspirantes se informen correctamente al tomar la decisión sobre qué carrera elegir para ellos.

Algunos de los beneficios de Difutech ante tales problemáticas son:

- Hacer llegar información sobre la oferta académica, actividades extraescolares y proyectos a los jóvenes que les interesa formar parte del ITSMT.
- Recopilar información para análisis de las preferencias de los jóvenes.
- Orientación a los jóvenes sobre qué carrera cursar.
- Informar notas relevantes a todos los que tienen la aplicación instalada.

La herramienta móvil Difutec puede dar al ITSMT la oportunidad de darse a conocer en muchos más lugares haciendo uso de tecnologías nuevas que están impactando en el desarrollo de aplicaciones móviles y de esta manera crecer como institución y ayudar para que más jóvenes puedan conocer sobre el ITSMT e ingresar a la institución, dando a la sociedad la oportunidad de tener más profesionistas preparados en algunas de las carreras ofertadas de la institución.

También es un beneficio para la institución ya que la hace crecer y entrar dentro del grupo de universidades que empiezan a implementar tecnologías Android sobre sus procesos y servicios de promoción y difusión.

1.5. Alcances y limitaciones

El presente trabajo se enfoca en el desarrollo de una App para dispositivos móviles que requiera la versión básica del sistema operativo Android (Versión 5). Dentro de las ventajas competitivas, se ofrece privacidad de la información que los usuarios o aspirantes proporcionen mediante el uso de la App. En la primera versión de la App se considera incluir la promoción y publicidad, sin embargo queda abierta la posibilidad de que la aplicación se pueda alojar en el Google Play debido a que esto dependerá de los recursos con los que se cuente para su alojamiento.

1.6. Organización del documento

Este trabajo se encuentra organizado en seis capítulos, con la intención de mostrar un panorama general del contenido del mismo, a continuación se describe brevemente el contenido de cada uno.

Capítulo I: Introducción.

Establece el planteamiento del problema e incluye escenarios que ejemplifiquen lo que se pretende trabajar, las hipótesis que presenta esta tesis, los objetivos generales y específicos, la justificación del porque se realiza este trabajo y la organización del mismo.

Capítulo II: Marco Teórico.

En este capítulo se presenta el marco teórico, todo aquel aspecto sobre las aplicaciones móviles, los tipos de aplicaciones móviles, el tema descriptivo de front-end y back-end, los lenguajes de programación para dispositivos móviles y los sistemas operativos de los mismos, que entornos de desarrollo existen para aplicaciones móviles y las bases de datos que estos utilizan.

Capítulo III: Estado del Arte.

En el capítulo III se presenta el estado del arte donde se incluirán todas aquellas investigaciones sobre aplicaciones similares que ya existen, sus objetivos, el impacto que han tenido y sus resultados.

Capítulo IV: Desarrollo e Implementación de la App.

En este capítulo se presentará el proceso de desarrollo e implementación de la aplicación, una vez que se aborden los temas teóricos y se expliquen las ventajas y desventajas de estos, se pondrán en uso las tecnologías que sean de utilidad para el desarrollo de la aplicación de acuerdo a los requerimientos que se solicitan, el diseño que se ha establecido en cuanto a pantallas y la base de datos.

Capítulo V: Resultados.

En este capítulo se ofrece la recolección, evaluación, discusión e interpretación de los resultados que surgieron durante el trabajo de investigación.

Capítulo VI: Conclusión.

Presenta una conclusión general del estudio además de presentar algunas líneas para ser tomadas en cuenta como trabajos futuros.

2 Marco Teórico

En este capítulo se aborda en principio que es una aplicación móvil, los puntos que se consideran al diseñarse las aplicaciones y que metodología puede guiar al proceso de desarrollo. Para poder seleccionar una metodología también es importante seleccionar el tipo de aplicación a programar por lo que se hace una mención de los tipos de aplicación que existen para seleccionar una. Dentro de lo que es el desarrollo de cualquier tipo de software de escritorio, web o de aplicación, también es importante separar la programación front-end y back-end, con motivo de tener un orden al realizar todos los pasos de la metodología y también al desarrollar la aplicación móvil, se hará una mención del significado de estos dos conceptos para ayudar a entender como se lleva a cabo la comunicación entre el usuario y la aplicación, aunque la información del front-end y back-end también ayuda a desarrollar una aplicación basada en los requerimientos del usuario y procesar la información de manera correcta convirtiéndola en información útil.

Toda la información requerida será de ayuda para guiar al desarrollo del proyecto aunque en sí, para tener los conocimientos que se necesita para desarrollarlo hace falta hacer mención de las teorías de todas las tecnologías que se requieren por ello también se hará mención de las herramientas que se necesitan como el lenguaje de programación de utilizar y el sistema operativo seleccionado como objetivo del proyecto, además de la base de datos que requiere y que cumpla con las características que se necesitan en la aplicación. Una vez seleccionadas esas tecnologías solo hará falta seleccionar a la herramienta IDE que será ideal considerando sus características, ventajas y desventajas, para construir la aplicación móvil. Las aplicaciones móviles hoy en día son consideradas como una herramienta indispensable en muchos ámbitos de la sociedad humana. Pueden tener múltiples utilidades o ser creadas para cumplir específicamente una sola tarea, sea cual sea el propósito de estas, su diseño y desarrollo se basa en metodologías que apenas son propuestas y también heredan de técnicas de programación orientadas a objetos o estructuradas.

2.1. Aplicaciones móviles

El acelerado desarrollo tecnológico que se ha presentado durante las últimas tres décadas ha propiciado que en particular las Tecnologías de Información y Comunicación (TIC) se han constituido como una herramienta indispensable para alcanzar mejores condiciones de bienestar y desarrollo entre las personas, empresas e instituciones. Un estudio de la UNESCO menciona que el crecimiento en el uso de dispositivos electrónicos como tabletas, teléfonos inteligentes, así como cualquier otro dispositivo móvil se está convirtiendo en una de las tendencias más importantes a nivel mundial.

Los dispositivos móviles ahora son una herramienta clave para múltiples tareas que se requieren cumplir todos los días en la vida de las personas. Pero, ¿Cómo las personas pueden cumplir esas actividades con estos dispositivos?

Imaginemos que una persona común tiene como destino su nuevo trabajo ubicado a una distancia desconocida de su hogar, la persona debe tener en cuenta el factor del tiempo para llegar puntual y el camino que debe tomar. En este caso hay dos tareas fundamentales que está realizando la persona: la primera es salir a tiempo para llegar a su destino y la segunda es tomar la ruta adecuada. Para que esta persona pueda ser auxiliada por su dispositivo móvil en estas tareas, el dispositivo móvil debería contar con una función de alarma para que pueda despertar a tiempo de poder llegar, también requiere que el dispositivo cuente con una función de mapa guiado para poder trazar una ruta hasta el destino. Estas funciones son lo que integran al dispositivo móvil para cumplir muchas tareas diarias de las personas, reciben el nombre de aplicaciones móviles o apps.[10]

Entonces ¿Cómo podemos definir a una aplicación? Resumiendo, que es una función dentro de un dispositivo móvil puede quedar de la siguiente manera: “Una aplicación (también llamada app) es simplemente un programa informático creado para llevar a cabo o facilitar una tarea en un dispositivo informático”.[12] Cabe destacar que, aunque todas las aplicaciones son programas, no todos los programas son aplicaciones. Existe multitud de software en el mercado, pero sólo se denomina así a aquel que ha sido creado con un fin determinado, para realizar tareas concretas. No se consideraría una aplicación, por ejemplo, un sistema operativo, ni una suite, pues su propósito es general.

Una aplicación móvil (App) es un pequeño programa que se puede descargar de la web

(en muchos casos tiendas on-line) para ser instalado en el teléfono smartphone, Tableta o reproductor MP3, a través de una conexión a Internet.

Las aplicaciones móviles son uno de los segmentos del marketing móvil que mayor crecimiento ha experimentado en los últimos años. Se pueden encontrar en la mayoría de los teléfonos, incluso en los modelos más básicos (donde proporcionan interfaces para el envío de mensajería o servicios de voz), aunque adquieren mayor relevancia en los nuevos teléfonos inteligentes.

Las apps se convierten, por lo tanto, en un soporte en sí mismo para comunicar, impactar, relacionarse, intercambiar y comerciar desde el punto de vista de marketing; ofrecen un importante servicio de valor añadido al usuario; además de ser en ocasiones soportes publicitarios para terceros (una de los formatos más efectivos en la comunicación publicitaria y de marketing). La app, al cubrir una necesidad de forma inmediata, independiente y personalizada, es el vehículo más directo para estrategias de marketing dirigido, usando criterios de segmentación exigentes.

El mundo de aplicaciones da un giro radical con la llegada de los smartphones y en especial con el lanzamiento del dispositivo iPhone a mediados de 2007. Con este lanzamiento, Apple cambia la manera de interactuar con el teléfono, convirtiéndolo en un dispositivo intuitivo, potente, táctil y siempre online. Apple también desarrolla en profundidad el concepto de “tienda de aplicaciones”: un mercado único y organizado donde la adquisición de las aplicaciones es transparente, fiable y directa. Además, incorpora un proceso definido y homogéneo de desarrollo para su sistema operativo: iOS, que ayuda a sacar el máximo potencial de las capacidades técnicas del teléfono, mejorando considerablemente la experiencia del usuario. Posteriormente, varios fabricantes, operadoras y empresas de software adoptan este modelo para ofrecer aplicaciones.

Hasta el momento, Google y su sistema operativo Android libre y de código abierto ha logrado posicionarse como principal competidor. Adopta también el modelo de “mercado de aplicaciones” y con la ventajosa diferencia de que cualquier fabricante (HTC, Sony Ericsson o Samsung, entre otros) puede elegir libremente “Android” como sistema operativo para sus teléfonos. [14]

2.2. Tipos de aplicaciones móviles

Muchas organizaciones que comienzan a implementar una estrategia móvil se enfrentan con una importante decisión que va a influir sobre los resultados de esta iniciativa. El proceso de elegir un enfoque de desarrollo para una aplicación móvil, ya sea nativa, Web o híbrida, implica muchos parámetros, como presupuesto, plazos del proyecto, destinatarios y funcionalidad de la aplicación, entre otros. Cada enfoque conlleva beneficios y limitaciones inherentes, y encontrar la más adecuada para las necesidades de la organización puede ser una tarea difícil. La finalidad de este documento no consiste en identificar el mejor enfoque de desarrollo, ya que no existe ninguno, sino enumerar los pros y los contras de cada uno y describir los distintos escenarios, o requisitos empresariales, que mejor se adecuan a cada uno de ellos.[15]

Aplicaciones nativas

Las aplicaciones nativas tienen archivos ejecutables binarios que se descargan directamente al dispositivo y se almacenan localmente. El proceso de instalación lo puede iniciar el usuario o, en algunos casos, el departamento de TI de la empresa. La manera más común de descargar una aplicación nativa es visitando una tienda de aplicaciones, como App Store de Apple, Marketplace de Android o App World de BlackBerry, pero existen otros métodos que a veces ofrece el proveedor móvil. Una vez que la aplicación ha sido instalada en el dispositivo, el usuario la ejecuta como cualquier otro servicio del dispositivo. Tras la inicialización, la aplicación nativa se conecta directamente con el sistema operativo móvil, sin ningún intermediario ni contenedor. La aplicación nativa puede acceder libremente a todas las APIs que el proveedor del SO ponga a disposición y, en muchos casos, tiene características y funciones únicas que son típicas de ese SO móvil en particular.

Aplicaciones móviles basadas en la Web

Los dispositivos móviles modernos cuentan con poderosos navegadores que dan soporte a muchas funcionalidades nuevas de HTML5, Cascading Style Sheets 3 (CSS3) y JavaScript de avanzada. Con los últimos avances logrados, HTML5 marca la transición de esta tecnología desde un “lenguaje de definición de páginas” a un poderoso estándar de desarrollo de aplicaciones complejas basadas en navegador.

Algunos ejemplos del potencial de HTML5 son: componentes IU avanzados, acceso a

múltiples tipos de medios, servicios de geoposicionamiento y disponibilidad offline. Al emplear estas características y muchas otras que aún se encuentran en desarrollo, los desarrolladores pueden crear aplicaciones avanzadas, usando únicamente tecnologías basadas en la Web.

Es útil distinguir entre dos enfoques radicalmente distintos de aplicaciones Web. Todos estamos familiarizados con la navegación móvil y los sitios Web optimizados para servicios móviles. Estos sitios reconocen cuando se accede a ellos a través de un smartphone y preparan páginas HTML que han sido diseñadas para brindar una “experiencia táctil” cómoda en una pantalla pequeña. Pero algunas empresas llegan aún más lejos y mejoran la experiencia del usuario creando un sitio Web móvil que se parece a una aplicación nativa y se puede ejecutar a partir de un acceso directo que es idéntico al que se utiliza para lanzar aplicaciones nativas.

Existe un amplio abanico de posibilidades entre esos dos extremos, ya que la mayoría de los sitios Web implementan su propio mix de características. Las aplicaciones móviles basadas en la Web constituyen una tendencia muy promisoría. Para sacar el mayor provecho de esta tendencia y ayudar a los desarrolladores a construir la UI para el cliente, se han creado cada vez más herramientas JavaScript, tales como `dojox.mobile`, `Sencha Touch` y `jQuery Mobile`, las cuales generan interfaces de usuario con una apariencia similar a las aplicaciones analíticas. Ambas se ejecutan completamente dentro del navegador del dispositivo móvil y utilizan las últimas características de JavaScript, CSS y HTML5 disponibles en los navegadores móviles modernos.

Aplicaciones híbridas.

El enfoque híbrido combina desarrollo nativo con tecnología Web. Usando este enfoque, los desarrolladores escriben gran parte de su aplicación en tecnologías Web para múltiples plataformas, y mantienen el acceso directo a APIs nativas cuando lo necesitan.

La porción nativa de la aplicación emplea APIs de sistemas operativos para crear un motor de búsqueda HTML incorporado que funcione como un puente entre el navegador y las APIs del dispositivo.

Este puente permite que la aplicación híbrida aproveche todas las características que ofrecen los dispositivos modernos.

Los desarrolladores de aplicaciones pueden optar por codificar su propio puente o bien aprovechar soluciones ya construidas, como `PhoneGap`, una biblioteca de fuente abier-

ta que provee una interfaz JavaScript uniforme para funcionalidades de dispositivos seleccionados que son iguales en todos los sistemas operativos.

La porción nativa de la aplicación se puede desarrollar independientemente, pero algunas soluciones del mercado ofrecen este tipo de contenedor nativo como parte de su producto, lo que brinda al desarrollador formas de crear una aplicación avanzada que utilice todas las funciones del dispositivo usando únicamente lenguajes Web. En algunos casos, una solución va a permitir que el desarrollador utilice cualquier conocimiento nativo que pueda tener para adaptar el contenedor nativo a las necesidades únicas de la organización.

La porción Web de la aplicación puede ser una página Web que resida en un servidor o bien un conjunto de archivos HTML, JavaScript, CSS y medios, incorporados en el código de la aplicación y almacenados localmente en el dispositivo. Ambos enfoques presentan ventajas y desventajas. El código HTML que está alojado en un servidor permite que los desarrolladores introduzcan pequeñas actualizaciones en la aplicación sin tener que seguir el proceso de entrega y aprobación que algunas tiendas de aplicaciones requieren.

Comparación de los distintos enfoques

A modo de resumen, en la Figura 2.1 se comparan los tres enfoques de desarrollo.

El enfoque nativo se destaca por su desempeño y acceso de los dispositivos, pero conlleva costos y requiere actualizaciones. El enfoque Web es mucho más simple, menos costoso y más fácil de actualizar, pero actualmente su funcionalidad es limitada y no puede alcanzar un alto nivel de experiencia del usuario como el de las llamadas API nativas.

El enfoque híbrido ofrece un término medio que, en muchas situaciones, constituye lo mejor de ambos mundos, en especial si el desarrollador desea emplearlo en múltiples sistemas operativos. En el Cuadro 2.1 se muestra una comparación entre los 3 enfoques.

Como se puede observar en el Cuadro 2.1, ninguno de los enfoques en sí mismo ofrece todos los beneficios todo el tiempo. Para elegir el enfoque más adecuado hay que tener en cuenta las necesidades específicas de la organización, y basarse en muchos parámetros, como presupuesto, plazos de entrega, recursos internos, mercado objetivo, funcionalidad requerida de la aplicación, infraestructura de TI, etc.



Figura 2.1: Comparación diseño básico entre aplicación nativa, basada en web y móvil. Fuente: [10]

Hay algo que es muy claro: La mayoría de las empresas actuales tienen que encontrar un punto medio, por un lado, entre la experiencia del usuario y la funcionalidad de las aplicaciones y, por el otro, entre los costos de desarrollo y el tiempo de salida al mercado. El desafío consiste en elegir el enfoque de desarrollo correcto que logre un equilibrio entre los requisitos de la organización con sus limitaciones vinculadas al presupuesto y al tiempo de salida al mercado.

2.3. Fases del diseño de aplicaciones móviles

Para el desarrollo de aplicaciones móviles, al igual que con el software que se construye para equipos de cómputo y software basado en la web, debe contener su metodología de desarrollo que permita al desarrollador tener una guía de los pasos que tiene que seguir para construir software para dispositivos móviles.

Múltiples propuestas se han presentado en los últimos años, la mayoría basadas en la

Característica	Aplicación nativa	Aplicación híbrida	Aplicación web
Lenguaje de desarrollo	Solo nativo	Nativo y Web o solo nativo	Solo web
Portabilidad y optimización de código	Bajo	Alto	Alto
Características de acceso	Alto	Mediano	Bajo
Uso de conocimiento existente	Bajo	Alto	Alto
Gráficos avanzados	Alto	Mediano	Mediano
Flexibilidad de actualizaciones	Bajo (siempre tiendas)	Mediano (con frecuencia tiendas)	Alto
Experiencia de instalación	Alta (a partir de la tienda)	Alta (a partir de la tienda)	Mediana (mediante el navegador móvil)

Cuadro 2.1: Tabla comparativa entre enfoque nativo, web e híbrido. Fuente:[10]

metodología propuesta por Roger S. Pressman, basándose en numerosas metodologías escribe que “*La existencia de un proceso del software no es garantía de que el software se entregue a tiempo, que satisfaga las necesidades de los consumidores o que tenga las características técnicas que conducirán a características de calidad a largo plazo*”.[18]

Se puede decir que todo desarrollador de software ha tenido experiencias con más de una metodología de desarrollo y ha sufrido fallas al construir un software. Por ello en el libro Ingeniería de Software se generalizo un solo proceso de desarrollo, pero con diferentes modelos adaptables a muchos tipos de software. En las últimas décadas se han propuesto numerosos enfoques para la evaluación y mejora de un proceso del software:

- Método de evaluación del estándar CMMI para el proceso de mejora (SCAMPI, por sus siglas en inglés): proporciona un modelo de cinco fases para evaluar el proceso: inicio, diagnóstico, establecimiento, actuación y aprendizaje. El método SCAMPI emplea el SEI CMMI como la base de la evaluación[21].

- Evaluación basada en CMM para la mejora del proceso interno (CBA IPI, por sus siglas en inglés): proporciona una técnica de diagnóstico para evaluar la madurez relativa de una organización de software; usa el SEI CMM como la base de la evaluación[5].
- SPICE (ISO/IEC 15504): estándar que define un conjunto de requerimientos para la evaluación del proceso del software. El objetivo del estándar es ayudar a las organizaciones a desarrollar una evaluación objetiva de cualquier proceso del software definido[18].
- ISO9001:2000 para software: estándar genérico que se aplica a cualquier organización que desee mejorar la calidad general de los productos, sistemas o servicios que proporciona. Por tanto, el estándar es directamente aplicable a las organizaciones y compañías de software[2].

Las metodologías ágiles para el desarrollo de software, surgieron como una solución inmediata, garantizando la realización de proyectos en corto, mediano y largo plazo, las cuales se pueden ajustar a los Time-to-Market de los servicios móviles. Sin embargo, no garantizan el potencial de éxito de estos, éxito que depende en gran parte del conocimiento electrónico y de telecomunicaciones, del entorno específico donde se ejecuta la aplicación (terminales y redes), además de la caracterización del cliente.

El desarrollo de aplicaciones para proveer servicios móviles, difiere del desarrollo de software tradicional en muchos aspectos, lo que provoca que las metodologías usadas para estos entornos móviles, también difieran de las del software clásico[20].

Las características especiales de los entornos móviles como el canal de radio, la capacidad de los terminales, la portabilidad, el tiempo de salida al mercado "Time-to-Market" la movilidad del usuario, entre otras; exigen nuevas tendencias para desarrollar el software móvil.

Proceso General de Software de Pressman

De acuerdo al proceso general para desarrollar software que se redacta en el Libro de Pressman, el proceso puede dividirse en 5 actividades[18]:

- **Comunicación.** Antes de que comience cualquier trabajo técnico, tiene importancia crítica comunicarse y colaborar con el cliente (y con otros participantes). Se busca entender los objetivos de los participantes respecto del proyecto, y re-

unir los requerimientos que ayuden a definir las características y funciones del software.

- **Planeación.** Cualquier viaje complicado se simplifica si existe un mapa. Un proyecto de software es un viaje difícil, y la actividad de planeación crea un “mapa” que guía al equipo mientras viaja. El mapa —llamado plan del proyecto de software— define el trabajo de ingeniería de software al describir las tareas técnicas por realizar, los riesgos probables, los recursos que se requieren, los productos del trabajo que se obtendrán y una programación de las actividades.
- **Modelado.** Ya sea usted diseñador de paisaje, constructor de puentes, ingeniero aeronáutico, carpintero o arquitecto, a diario trabaja con modelos. Crea un “bosquejo” del objeto por hacer a fin de entender el panorama general —cómo se verá arquitectónicamente, cómo ajustan entre sí las partes constituyentes y muchas características más—. Si se requiere, refina el bosquejo con más y más detalles en un esfuerzo por comprender mejor el problema y cómo resolverlo. Un ingeniero de software hace lo mismo al crear modelos a fin de entender mejor los requerimientos del software y el diseño que los satisfará.
- **Construcción.** Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores en éste.
- **Despliegue.** El software (como entidad completa o como un incremento parcialmente terminado) se entrega al consumidor que lo evalúa y que le da retroalimentación, misma que se basa en dicha evaluación.

Para muchos proyectos de software, las actividades estructurales se aplican en forma iterativa a medida que avanza el proyecto. Es decir, la comunicación, la planeación, el modelado, la construcción y el despliegue se ejecutan a través de cierto número de repeticiones del proyecto. Cada iteración produce un incremento del software que da a los participantes un subconjunto de características y funcionalidad generales del software. Conforme se produce cada incremento, el software se hace más y más completo.

Metodología para el Desarrollo de Aplicaciones Móviles MDAM

Esta metodología propuesta en la Universidad Distrital Francisco José de Caldas en Colombia para el desarrollo de aplicaciones para móviles se fundamenta en la experiencia de investigaciones previas en aplicaciones móviles, la evaluación del potencial de éxito para servicios de tercera generación denominada 6 M, la ingeniería de software

educativo con modelado orientado por objetos, y principalmente en los valores de las metodologías ágiles.

De las 6 M's se extrae la concepción de que las aplicaciones móviles deben garantizar el cumplimiento de las necesidades de los usuarios y al mismo tiempo generen ingresos. La 6 M's debe su nombre a los seis atributos que se miden para evaluar el éxito del servicio propuesto: Movement (Movimiento), Moment (Momento), Me (Yo), Multi-user (Multiusuario), Money (Dinero) y Machines (Máquinas)[1].

De acuerdo con las necesidades a cubrir, en esta metodología se establecen 5 fases[8]:

- **Análisis.** En esta fase se analizan las peticiones o requerimientos de las personas o entidad para la cual se desarrolla el servicio móvil "Cliente", el propósito es definir las características del mundo o entorno de la aplicación.
- **Diseño.** El objetivo de esta etapa es plasmar el pensamiento de la solución mediante diagramas o esquemas, considerando la mejor alternativa al integrar aspectos técnicos, funcionales, sociales y económicos. A esta fase se retorna si no se obtiene lo deseado en la etapa prueba de funcionamiento.
- **Desarrollo.** El objetivo de esta fase es implementar el diseño en un producto de software.
- **Pruebas de funcionamiento.** El objetivo de esta fase es verificar el funcionamiento de la aplicación en diferentes escenarios y condiciones.
- **Entrega.** Terminada la depuración de la aplicación y atendidos todos los requerimientos de última hora del cliente se da por finalizada la aplicación y se procede a la entrega del ejecutable, el código fuente, la documentación y el manual del sistema.

Al igual que el proceso general de software de Pressman este también es iterativo, es decir, se puede regresar a una fase previa en caso de alguna equivocación. Aunque también consta de 5 fases existe la diferencia de que en el proceso general de software de Pressman la fase de pruebas está realizada en la etapa de despliegue, ya que al ser iterativo permite por cada iteración un incremento o mejora al software mientras que la metodología MDAM contiene su propia fase de pruebas para evaluación del funcionamiento correcto del software móvil.

Metodología de Desarrollo Ágil de Aplicaciones Android (D3A)

Esta metodología de desarrollo para aplicaciones android se basa en el desarrollo de software móvil para empresas de tamaño PYME (Pequeñas Y Medianas Empresas). Este tipo de empresas suele tener un alcance local y regional por su cantidad de empleados y su principal característica, el crecimiento que se les puede permitir, ya que una gran empresa puede impactar de manera internacional mas no crecer, pero una empresa pyme si es capaz de impactar y crecer exponencialmente, este crecimiento puede darse gracias a la implementación de un software móvil, con este se pueden mejorar los procesos de básicos de una pyme y llegar a más personas de la región.

La mayoría de los proyectos de software que se desarrollan en PYMES tienen un elevado porcentaje de fracaso, éste puede ser completo, es decir que no se lleve a cabo la entrega del producto ó parcial, es decir, que no se cumpla con los requerimientos solicitados por el cliente, o que se tengan problemas fuertes de errores en la implementación, dificultad para el mantenimiento y desfase en atención a las necesidades.

Las tres variables más relevantes en el fracaso de los proyectos de software son a) Errores de estimación de Costo; b) Falla en la medición del riesgo del proyecto, y c) Desarrollo basado en programadores estrella.

Como mecanismo para paliar lo anterior, el uso de metodologías de desarrollo de software ha demostrado ser una excelente práctica que contribuye al cumplimiento de fechas de entrega, al mismo tiempo que asegura la calidad del software y atiende las necesidades del cliente. En especial las metodologías ágiles permiten a las PYMES de desarrollo de software responder de manera eficiente a los cambios en los requerimientos del producto en desarrollo, a la vez que mantienen el control de riesgo, sin caer en exceso de mecanismos de control.

Como respuesta a las tres variables en el fracaso de proyectos de software, se hizo un análisis y evaluación, concluyendo en una metodología de 7 etapas[17]:

- Recolección de requerimientos. Proceso de análisis de requerimientos con las técnicas de entrevista con el cliente y el establecimiento de acuerdos para realizar un desarrollo gradual y basado en prototipos.
- Diseño rápido. Se realiza una comparativa de herramientas de prototipado rápido para el diseño de GUI, así mismo, para el caso de usabilidad y desarrollo de interfaces de usuario se plantearon dos vertientes, la primera radica en el diseño

de interfaces basado en estándares comunes como la teoría de colores y la segunda radica en utilizar los estándares que Google como empresa fundadora del sistema nos recomienda.

- Definición de tareas. Es la definición de tareas se desarrollaba mediante un tablero Scrum.
- Construcción del prototipo (interfaces y código). Se diseña la arquitectura de la aplicación, diseño de las interfaces y selección del IDE para la captura del código.
- Evaluación del prototipo por el representante de proyecto. El representante del proyecto selecciona un grupo de “pruebas falla” para medir la resistencia de la aplicación contra mal uso del usuario.
- Evaluación del prototipo por el cliente. Realiza un grupo de pruebas funcionales basadas en las características de la aplicación.
- Lanzamiento. Se lanza la versión desarrollada durante la metodología.

Metodología a utilizar

En base al conocimiento que se toma de la metodología propuesta por pressman se agregan las siguientes etapas de la metodología propuesta por Ponce Mendoza Ulises quedando de la siguiente manera la metodología a utilizar:

1. Recolección y análisis de requerimientos. En esta etapa se enlistarán los requerimientos básicos y funciones principales que requiere el cliente para la aplicación.
2. Diseño rápido. En esta etapa se realizará un diseño de la estructura de las pantallas de acuerdo a la funcionalidad y a los requisitos solicitados.
3. Planeación. En esta etapa se realizará de acuerdo al diseño rápido y el análisis de requerimientos cual será el orden de elaboración de los módulos de la aplicación.
4. Modelado. En esta etapa se van a modelar las interfaces que se van a requerir en base a las estructuras creadas en la etapa del diseño rápido.
5. Construcción de Prototipo. En esta etapa se va a generar el código para que las interfaces puedan funcionar y cumplir sus objetivos.
6. Pruebas y Evaluaciones. En esta etapa se va a evaluar que los módulos funcionen correctamente de acuerdo a sus objetivos y puedan satisfacer también las necesidades del cliente.

7. Lanzamiento. Después de ser aprobada totalmente la aplicación previa a su lanzamiento se construye su manual de uso y se libera en totalidad la aplicación.

La metodología bajo la que se construye la aplicación móvil DIFUTEC es una metodología desarrollada en base a dos distintas por lo que se pueden considerar características positivas.

Características:

La metodología al ser elegida en base a la de Pressman puede ser iterativa es decir puede cumplir con el software en proporciones pequeñas hasta su entrega total por lo que puede considerarse que esta metodología contiene la esencia de un proceso ágil, puesto que su desarrollo es incremental, otra característica es que esta metodología es cooperativa, ya que clientes y desarrolladores trabajan juntos constantemente para obtener el mejor producto de software, debe ser un método sencillo, sus etapas no deben ser confundibles para que pueda ser fácil de aprender y modificar algunas actividades de este, y como última característica es que es adaptable ya que permite involucrarse a cualquier desarrollo de otro tipo de software de aplicación.

2.4. Front-end y Back-end

En el diseño de cualquier tipo de software (aplicación, escritorio y web) es necesario siempre estar concientes de las partes se comunicaran con el usuario y las partes que reciben, interpretan y transforman los datos en información útil.

Para poder dividir un software en cuanto a comunicación con el usuario y funcionalidad es necesario conocer los conceptos de front-end y back-end.

Front-end

En el desarrollo del software el front-end es la parte que interactúa con el o los usuarios. De una manera práctica, el front-end es todos los formularios o ventanas que el usuario puede visualizar directamente al ingresar a una página web, a un programa de escritorio o en una pantalla de alguna aplicación.

Back-end

En el desarrollo del software back-end es la parte que procesa la entrada de datos realizado por los usuarios desde la parte del front-end es decir, son los procesos que

efectúan los respectivos sistemas para resolver las peticiones de los usuarios.

De esta manera conjunta el front-end y el back-end interactúan en un software para resolver las necesidades de los usuarios.

La idea de esta abstracción es mantener separadas las diferentes partes de un sistema web o software con el fin de tener un mejor control. En pocas palabras el objetivo es que el front-end recopile los datos y el back-end los procese.[13]

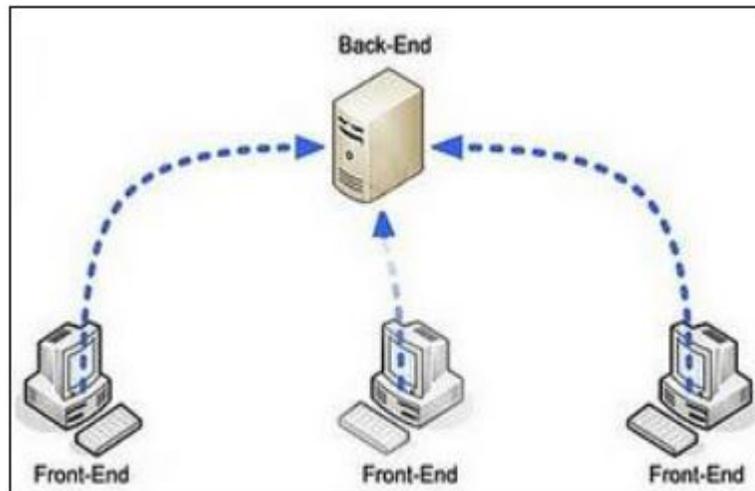


Figura 2.2: Diagrama interpretativo del front-end y back-end. Fuente: [13]

2.5. Lenguajes de programación para dispositivos móviles

Los lenguajes de programación para dispositivos móviles actualmente tienen mucha demanda en cuanto a conocimiento por la alta demanda de desarrollo de aplicaciones para diversos sistemas operativos que hay para dispositivos móviles. Dentro de los principales sistemas operativos para dispositivos móviles podemos encontrar a los 2 fabricantes informáticos con más demanda en el mercado de dispositivos móviles, estos 2 sistemas operativos son:

1. Android.
2. IOS.

Para cada uno de ellos existen lenguajes de programación nativos y tienen características que los hace muy competitivos para cada aplicación que intenta desarrollarse

diariamente. Entonces para que un lenguaje de programación pueda considerarse competitivo debe tener buenas características y soporte. A continuación se hará una descripción de los 3 principales lenguajes de programación para dispositivos móviles.[16]

Java (C, C++)

Java es uno de los lenguajes de programación más populares del mundo. Es un lenguaje orientado a objetos, potente, versátil y multiplataforma (corre en cualquier sistema operativo moderno). Además puedes obtener Java y gran cantidad de herramientas para trabajar con él de forma gratuita, siendo la mayor parte de su código libre y abierto. Java fue elegido como el lenguaje para el entorno de desarrollo de Android, el sistema operativo móvil líder en smartphones y tablets. Android es por tanto el sistema operativo (es una versión de Linux) y Java el lenguaje utilizado para crear apps en él.

La principal característica -y ventaja- de este lenguaje de programación es que se trata de un lenguaje independiente de la plataforma, es decir, cualquier programa creado a través de Java podrá funcionar correctamente en ordenadores de todo tipo y con sistemas operativos distintos siempre que se cumpla, que el equipo tenga instalada la máquina virtual de Java (JVM).

Java es, por otra parte, un lenguaje que se adapta a la perfección a todo tipo de dispositivos móviles como las incipientes tablets o smartphones, una ventaja que permite que cualquier desarrollo de software creado a través de Java sea visible desde cualquier lugar y en cualquier momento.

Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, enlace dinámico y polimorfismo. Los modelos de objetos son llamados, como en C++, clases y sus copias, instancias. Estas instancias, como en C++, necesitan ser construidas y destruidas en espacios de memoria.[24]

Java incorpora funcionalidades inexistentes en C++ como por ejemplo, la resolución dinámica de métodos. Esta característica deriva del lenguaje ObjectiveC. En C++ se suele trabajar con librerías dinámicas (DLLs) que obligan a recompilar la aplicación cuando se retocan las funciones que se encuentran en su interior. Java resuelve este inconveniente mediante una interfaz específica llamada RTTI (Run Time Type

Identification) que define la interacción entre objetos excluyendo variables de instancias o implementación de métodos. Las clases en Java tienen una representación en el runtime que permite a los programadores interrogar por el tipo de clase y enlazar dinámicamente la clase con el resultado de la búsqueda.

Características de Java[25]:

1. Lenguaje Simple: Se lo conoce como lenguaje simple por que viene de la misma estructura de C y C++; ya que C++ fue un referente para la creación de java por eso utiliza determinadas características de C++ y se han eliminado otras.
2. Orientado a Objetos: Toda la programación en java en su mayoría está orientada a objetos, ya que al estar agrupados en estructuras encapsuladas es más fácil su manipulación.
3. Robusto: Es altamente fiable en comparación con C, se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.
4. Seguro: La seguridad es una característica muy importante en java ya que se han implementado barreras de seguridad en el lenguaje y en el sistema de ejecución de tiempo real.
5. Portable: Por ser indiferente a la arquitectura sobre la cual está trabajando, esto hace que su portabilidad sea muy eficiente, sus programas son iguales en cualquiera de las plataformas, ya que java especifica tamaños básicos, esto se conoce como la máquina virtual de Java.
6. Interpretado y compilado a la vez: Java puede ser compilado e interpretado en tiempo real, ya que cuando se construye el código fuente este se transforma en una especie de código de máquina.
7. Multihebra o Multihilos: Java tiene una facilidad de cumplir varias funciones al mismo tiempo, gracias a su función de multihilos ya que por cada hilo que el programa tenga se ejecutaran en tiempo real muchas funciones al mismo tiempo.
8. Alto rendimiento: Java es considerado de alto rendimiento por ser tan veloz en el momento de correr los programas y por ahorrarse muchas líneas de código.

Objective - C

Objective-C es una extensión orientada a objetos de la sintaxis del lenguaje de programación C, utilizado en Mac OS X y GNUStep. Para desarrollar aplicaciones con este lenguaje, es recomendable saber programar en C, y si además sabes programar en C++, Java o C#, será más fácil. Es importante tener presente que en iPhone no existe recolector de basura, por lo que es recomendable estudiar en profundidad la documentación oficial de Apple sobre Objective-C y la gestión de memoria, aunque en la versión Objective-C 2.0 se incluye un recolector de basura, este no es soportado por iPhone, por lo que cada instancia tiene un contador de referencias y cuando llega a cero el objeto es liberado de la memoria. Es decir, que los objetos de la API oficial de Apple a los que les pasemos objetos creados por nosotros, para almacenarlos en su interior, los retendrá aumentando el contador y los liberará cuando el objeto sea destruido.[26]

Objective-C comparte elementos comunes con el lenguaje C como: sentencias de control de flujo (if, for, while, etc.); tipos de datos fundamentales; estructuras y punteros; conversiones implícitas y explícitas entre tipos; el ámbito de las variables (globales, estáticas o locales); las funciones y su sintaxis; las directivas del preprocesador (añadiendo Objective-C las suyas, así como las llamadas directivas del compilador).

Las clases en Objective-C se crean mediante dos archivos: un .h para definirla y un .m para implementarla. Por tanto, hay que utilizar `#import` para decirle al compilador las clases que vamos a utilizar en nuestro código. Se puede utilizar los mismos tipos que en C, más lo que añade el propio lenguaje.

En Objective-C se aplican los enlaces dinámicos donde las variables son identificadas por el runtime y no por el compilador y solo si existen punteros de objetos. De esta forma, comparando los lenguajes C y C++ con Objective-C veremos que los métodos y funciones deben ser llamados en alguna parte del código y se comprueba que efectivamente funcionen (en C++ por parte del compilador). En Objective-C el programa compila todos los datos y los objetos son los que deciden cuándo ejecutar los métodos.

Es un lenguaje interpretativo que permite crear clases, métodos y funciones, pero su sintaxis cambia al ser más dinámico. El lenguaje Objective-C es débilmente tipado, al contrario que C++, que es un lenguaje mayormente tipado y debe realizar conversiones de lenguaje para que sea más ordenado por defecto, pero al igual que el lenguaje C. Esto significa que una variable puede tener muchos valores distintos, por ejemplo, valores no específicos, cambiantes, que aunque estén relacionados no son los mismos.

Características de Objective - C:

1. Orientado a objetos: Es un lenguaje orientado a objetos basado altamente dinámico para su manipulación.
2. Mensajes: Los objetos en Objective-C son extremadamente autónomos, no son contenedores pasivos de comportamientos y estados, ni son datos y una colección de funciones que pueden ser aplicadas a los datos. Objective-C reafirma este concepto, permitiendo a cualquier mensaje- una solicitud para desarrollar una acción en particular- ser pasado a cualquier objeto.
3. Compilado: Objective-C es un lenguaje compilado. pero también es al mismo tiempo un lenguaje enlazado . Esto quiere decir básicamente que el resultado del compilador no es el programa final sino que existe una segunda fase que lleva a cabo el enlazador (linker en inglés).
4. Memoria dinámica. Objective-C crea sus objetos siempre en memoria dinámica para darles menos peso y favorecer la rapidez del proceso, contrario a lo que sucede con los lenguajes C y C++, los cuales deben adaptarse a la memoria dinámica y tienen un mayor peso.
5. Tipos dinámicos. El tipado en Objective-C es mucho más amplio que en C++. nn Introspección. Se utiliza para preguntar acerca de los datos y su uso, por ejemplo, cuando se requiere información sobre a qué clase u objeto pertenece tal variable.
6. Enlace dinámico. Objective-C busca en el propio objeto y clase la función de ejecución, a diferencia, por ejemplo, del lenguaje C++ donde en el compilador se decide qué funciones realizar para el objeto.
7. Carga dinámica. Cuando se cargan las funciones del programa, el lenguaje busca nuevas formas de actualizarse y completar lo que le falta (por ejemplo, a través de plugins).

2.6. Sistemas operativos para dispositivos móviles

Actualmente en la sociedad de muchos países con acceso a dispositivos móviles inteligentes, se posicionan muchas marcas de smartphones hasta arriba como los gigantes en ventas, esta popularidad no solo viene de las características del dispositivo sino del

sistema operativo que contiene. Poniendo como ejemplo marcas como Samsung, LG, Huawei, HTC, Sony y Motorola lanzan modelos de equipos de manera muy constante adaptándolos para ser compatibles con el sistema operativo Android de Google, por su contraparte, Apple, único fabricante de los iPhone utiliza únicamente su propio sistema operativo IOs.[16]

Para un desarrollador de aplicaciones móviles nativas es necesario elegir un sistema operativo sobre el cual trabajar, ya que una aplicación nativa requiere que se programe para una sola plataforma y aunque ya existen tecnologías como Xamarin que nos permite a través de C# programar aplicaciones para diferentes sistemas operativos, siempre es necesario conocer los sistemas para los cuales se desea realizar una aplicación. Entonces es necesario entender la siguiente pregunta:

¿Qué es un sistema operativo para dispositivo móvil?

Un sistema operativo móvil es un programa (software) que se inicia al encender el ordenador o el teléfono móvil y se encarga de gestionar todos los recursos del sistema informático, tanto de hardware (partes físicas, disco duro, almacenamiento, pantalla, teclado, etc.) como el software (programas e instrucciones) permitiendo así la comunicación entre el usuario y el ordenador. En definitiva controlan el ordenador, el teléfono móvil o la tablet y nos permite comunicarnos con ellos de forma sencilla.

Al igual que Linux o Windows son sistemas operativos que controlan su escritorio del PC o del portátil, un sistema operativo móvil es la plataforma de software sobre la que otros programas puedan ejecutarse en dispositivos móviles.

Los sistemas operativos móviles para los Smartphone son bastantes más simples y menos robustos que los de los PC y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos.

2.6.1. Sistemas operativos móviles principales

Hay muchos sistemas operativos para dispositivos móviles, como se mencionó anteriormente estos se eligen por el fabricante del dispositivo ya que lo adaptan para el SO seleccionado y así trabajar en mejorar la experiencia del usuario en comunicación con el smartphone, los principales sistemas operativos para dispositivos móviles actualmente son: Android y IOs.

2.6.2. Android

El lanzamiento de Android como plataforma para el desarrollo de aplicaciones móviles ha causado una gran expectación y está teniendo una importante aceptación, tanto por los usuarios como por la industria. En la actualidad es la alternativa más utilizada y aceptada frente a otras plataformas como iPhone.

Android es una plataforma formada por un conjunto de software en estructura de pila (software stack) que incluye un sistema operativo, software para conectar aplicaciones (middleware) y aplicaciones base. El SDK (Software Development Kit, Kit de Desarrollo de Software) de Android proporciona varias herramientas y API (Applications Programming Interface, Interfaz de Programación de Aplicaciones) que son necesarias para desarrollar aplicaciones Android. Estas aplicaciones se desarrollan en lenguaje Java.[16]

Android presenta una serie de características que lo hacen diferente. Es el primero que combina, en una misma solución, las siguientes cualidades[9]:

1. Plataforma realmente abierta. Es una plataforma de desarrollo libre basada en Linux y de código abierto. Una de sus grandes ventajas es que se puede usar y “customizar” el sistema sin pagar royalties.
2. Arquitectura basada en componentes inspirados en Internet. Por ejemplo, el diseño de la interfaz de usuario se hace en XML (eXtensible Markup Language), lo que permite que una misma aplicación se ejecute en un móvil de pantalla reducida o en un netbook.
3. Filosofía de dispositivo siempre conectado a Internet.
4. Gran cantidad de servicios incorporados. Por ejemplo, localización basada tanto en GPS (Global Positioning System) como en redes, bases de datos con SQL(Structured Query Language), reconocimiento y síntesis de voz, navegador, multimedia, etc.
5. Aceptable nivel de seguridad. Los programas se encuentran aislados unos de otros gracias al concepto de ejecución dentro de una caja que hereda de Linux. Además, cada aplicación dispone de una serie de permisos que limitan su rango de actuación (servicios de localización, acceso a Internet, etc.).
6. Optimizado para baja potencia y poca memoria. Por ejemplo, Android utiliza

la Máquina Virtual Dalvik. Se trata de una implementación de Google de la máquina virtual de Java optimizada para dispositivos móviles.

2.6.2.1. Arquitectura de Android

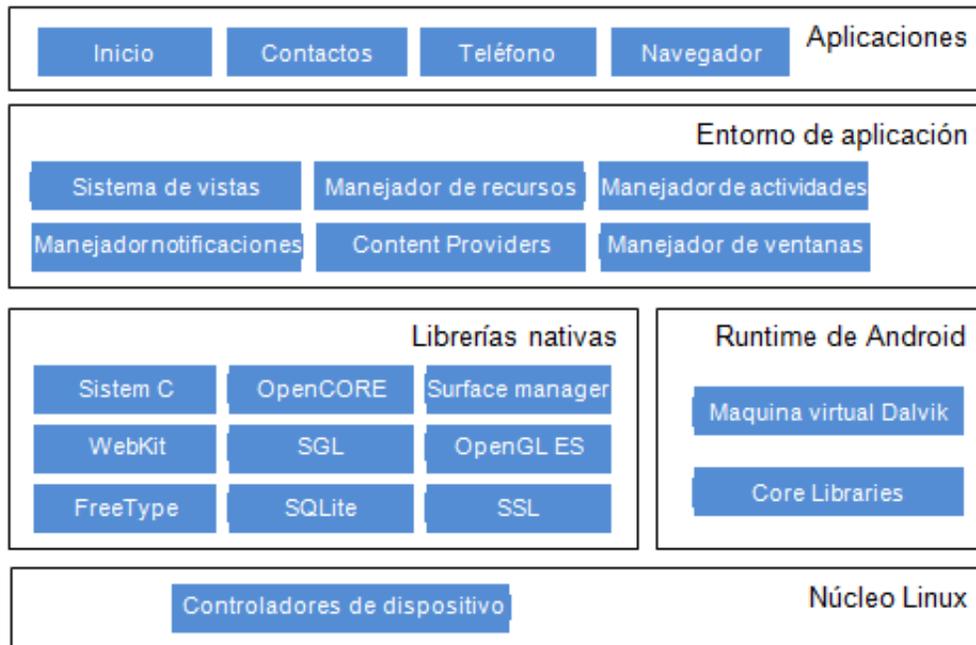


Figura 2.3: Arquitectura de Android. Fuente: [9]

Núcleo Linux

El núcleo de Android está formado por el sistema operativo Linux, versión 2.6. Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos. Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila. Por lo tanto, es la única que es dependiente del hardware.

Runtime de Android

Está basado en el concepto de máquina virtual utilizado en Java. Dado las limitaciones de los dispositivos donde ha de ejecutarse Android (poca memoria y procesador limitado) no fue posible utilizar una máquina virtual Java estándar. Google tomó la

decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones.

Librerías Nativas

Incluye un conjunto de librerías en C/C++ usadas en varios componentes de Android. Están compiladas en el código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son[16]:

- System C library: una derivación de la librería BSD de C estándar (libe), adaptada para dispositivos embebidos basados en Linux.
- Media Framework: librería basada en PacketVideo's OpenCORE; soporta codees de reproducción y grabación de multitud de formatos de audio, vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- Surface Manager: maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- WebKit: soporta un moderno navegador web utilizado en el navegador Android y en la vista webview. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.
- SQLite potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.

Entorno de aplicación

Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones (sensores, localización, servicios, barra de notificaciones, etc.). Esta capa ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas (sujetas a las restricciones de seguridad). Este mismo mecanismo permite a los usuarios reemplazar componentes. Una de las mayores fortalezas del entorno de aplicación de Android es que se aprovecha el lenguaje de programación Java. El SDK de Android no acaba de ofrecer todo lo disponible para su estándar del entorno de ejecución Java (JRE), pero es compatible con una fracción muy significativa de la misma. Los servicios más importantes que incluye son:

- Views: extenso conjunto de vistas (parte visual de los componentes).

- Resource Manager: proporciona acceso a recursos que no son en el código.
- Activity Manager: maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
- Notification Manager: permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
- Content Providers: mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos).

Aplicaciones

Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de ser ejecutadas en la máquina virtual Dalvik para garantizar la seguridad del sistema. Normalmente, las aplicaciones Android están escritas en Java. Para desarrollar aplicaciones en Java podemos utilizar el Android SDK. Existe otra opción consistente en desarrollar las aplicaciones utilizando C/C++. Para esta opción podemos utilizar el Android NDK (Native Development Kit).

2.6.3. IOs

A mediados de 2007 la tecnología Apple nos ofreció iOS (inicialmente llamado iPhone OS), desarrollado originalmente para el iPhone y con él, una nueva definición del teléfono móvil. Más tarde fue introducido en el iPod Touch y actualmente en el iPad.

El primer sistema operativo IOS lanzado en el año 2007 junto con el primer iPhone supuso toda una revolución en el mundo de la telefonía móvil ya que fue el primer teléfono móvil con pantalla multitouch, que permite al usuario utilizar varios puntos para interactuar con la pantalla y no únicamente un punto como hasta ese momento. También creo tendencia en cuanto a usabilidad y mejora de la experiencia del usuario en el uso de teléfonos móviles inteligentes.

2.6.3.1. Arquitectura de IOs

La arquitectura iOS que se muestra en la Figura 2.4 se basa en capas, donde las capas más altas contienen los servicios y las tecnologías indispensables para el desarrollo de aplicaciones y las capas más bajas son las encargadas de controlar los servicios básicos.[16]

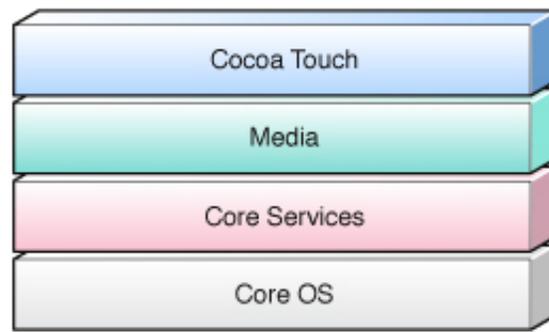


Figura 2.4: Arquitectura de IOs. Fuente: [26]

Cocoa Touch

Es la capa superior, la que los usuarios utilizan para interactuar con las aplicaciones, es decir, la capa visible. Es la zona donde nos encontramos los componentes visuales, se trata de una capa de abstracción. Posee tres frameworks:

1. UIKit, contiene todas las clases necesarias para el desarrollo de una interfaz de usuario.
2. Foundation Framework, define las clases básicas de acceso, manejo de objetos.
3. Servicios del S.O. Estas frameworks proporcionan la API de Cocoa para desarrollar aplicaciones.

Media

Esta capa proporciona los servicios gráficos y multimedia a la capa superior. Se trata de una capa basada en la mezcla de lenguaje C y Objective C que contiene las tecnologías que dan acceso a ficheros multimedia relacionados con audio, gráficos, vídeos, etc.

Core Services

Esta capa proporciona los servicios fundamentales del sistema utilizados por todas las aplicaciones. Se trata de la capa de servicios principales disponibles en el dispositivo y que pueden ser utilizados por todas las aplicaciones, como pueden ser: base de datos SQLite, acceso a la red, soporte para XML.

Core OS

En esta capa se encuentran servicios de bajo nivel como los archivos de memoria, manejo de memoria, seguridad y drivers del dispositivo. El sistema operativo iOS está basado en el OS X de Apple, que fue desarrollado a partir de una base Unix. Elementos de seguridad, memoria, procesos o manejo de ficheros son los que podemos encontrar en esta capa.

2.7. Entorno de desarrollo para aplicaciones móviles Android

Siendo Android el sistema operativo más utilizado en la actualidad, es interesante pensar en una idea que poder materializar y que sea parte del mundo que rodea al smartphone. Y al utilizar el lenguaje Java para desarrollar esa idea, lleva como responsabilidad seleccionar el entorno a escoger como herramienta para desarrollo. Esta herramienta se conoce como IDE (Integrated Development Environment) por sus siglas en inglés, significa Entorno de Desarrollo Integrado. Estos IDE's son herramientas que complementan el proceso de desarrollo de cualquier aplicación nativa, después de seleccionar el sistema operativo y el lenguaje de programación, sigue seleccionar la herramienta que ayude en el proceso de desarrollo de la aplicación. Para que un IDE sea competitivo y aceptable para el desarrollo de una aplicación móvil debe como mínimo cumplir las siguientes características:

- Editor de código
- Compilador
- Depurador (debbuger)
- Constructor de interfaz gráfica

El desarrollo de aplicaciones Android se realiza con un grupo de herramientas que son suministradas en el SDK. La utilización de este grupo de herramientas puede ser de dos formas: utilizando un Entorno de Desarrollo Integrado (IDE) en combinación con un plugin llamado ADT (Android Development Tools, Herramientas de Desarrollo para Android).

La manera más conocidas de desarrollar una aplicación para Android es utilizando como IDE a Android Studio o Eclipse

2.7.1. Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA[28]. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android, como las siguientes:

- Un sistema de compilación basado en Gradle flexible.
- Un emulador rápido con varias funciones.
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos.
- Integración de plantillas de código y GitHub para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código.
- Gran cantidad de herramientas y frameworks de prueba.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
- Compatibilidad con C++ y NDK

2.7.2. Eclipse

Eclipse es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). El entorno Eclipse se puede descargar desde el sitio web de descargas de Eclipse. La versión Eclipse IDE for Java Developers es una buena opción.

- Perspectivas, editores y vistas.
- Gestión de proyectos.

- Depurador de código.
- Extensa colección de plug-ins.

2.8. Bases de datos para dispositivos móviles

En el comienzo de los dispositivos móviles (Palm, IPAQ, PDA'S) sus aplicaciones inalámbricas eran totalmente desconectadas de las empresas o sistemas de computación. En otras palabras no requerían el intercambio de información con otros sistemas o con uno centralizado. Así que para aquel entonces no se hablaba de tiempo real en gestión de datos a nivel móvil. Las aplicaciones más reconocidas se limitaban a libreta de direcciones, horarios, organizadores, juegos, agendas y las más sofisticadas contaban con un pequeño paquete de oficina. Algunas de esas aplicaciones contaban con pequeñas bases de datos y otras almacenaban la información en archivos de texto; para aquel momento funcionaban correctamente, pero a medida que avanza la tecnología, las personas necesitan más. Por eso los fabricantes de SGBD (Sistemas Gestores de Bases de Datos) se vieron en la necesidad de crear SGBD móviles que permitan la construcción de bases de datos relacionales para los dispositivos móviles que tal cual se denominan Bases de Datos Móviles.

En los últimos años, la cantidad de dispositivos móviles informáticos se ha incrementado exponencialmente a nivel laboral, educativo y personal, ya que estos dispositivos permiten a los usuarios interactuar de la misma manera que lo hacen con equipos de hogar o de oficina pero desde lugares remotos. Estos dispositivos móviles (celulares, PDAs, computadores portátiles entre otros) usan cada día las nuevas tecnologías inalámbricas que ofrece el mercado y que dan un mayor cubrimiento en cuanto a la red. No obstante, el mayor problema que limita estos desarrollos tecnológicos, es que los sistemas de información se ven afectados por problemas de vulnerabilidad en la seguridad de sus datos, así como su costo y la disponibilidad en cualquier momento de esta información hacia el usuario.

Ante tal situación, las bases de datos móviles pueden ofrecer soluciones a algunos de los aspectos mencionados con anterioridad. Una base de datos móvil es aquella que es portable y posee una independencia del servidor corporativo de bases de datos, pero puede comunicarse con este servidor desde cualquier punto remoto para compartir

datos corporativos. La arquitectura básica de una base de datos móvil se muestra en la Figura 2.5.

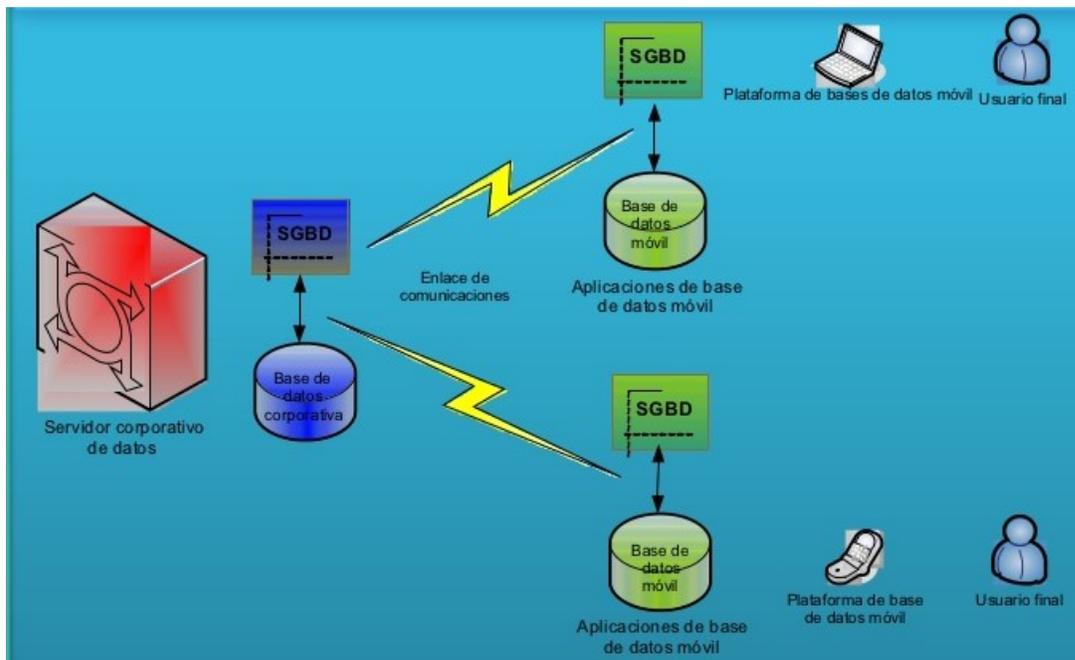


Figura 2.5: Arquitectura de base de datos móvil. Fuente [30]

2.8.1. Ejemplos de bases de datos para dispositivos móviles

Algunos de los Sistemas Gestores de Bases de Datos para dispositivos Móviles son los siguientes:

SQLite[7]

- SQLite es un motor de base de datos móvil de open source.
- Es un motor de base de datos sin servidor que necesita sin necesidad de configuración.
- SQLite es una opción popular como base de datos móvil para almacenamiento local en aplicaciones móviles.
- El motor SQLite no tiene procesos independientes con los que el programa de aplicación se comunica.

- SQLite implementa la mayor parte del estándar SQL-92.

IBM DB2 Everyplace[10]

- DB2e ha sido descontinuado y abril de 2013 es el final de la fecha de soporte.
- Tenía la mayor cuota de mercado después de SQL Anywhere.
- Tenía la huella de memoria más pequeña (350 KB) en comparación con otras bases de datos móviles comerciales.
- IBM ha reemplazado DB2e con la familia IBM solidDB.
- SolidDB es un MDBMS en memoria con características robustas de captura de datos.

Firestore Database[31]

- Firestore Database es una herramienta de base de datos de tipo NoSQL que forma parte de las tecnologías de desarrollo de Google.
- Posee una base de datos por nodos con una profundidad de 36 niveles.
- Contiene sus propias funciones de consulta a la base de datos.
- No requiere de un servidor propio para ser configurado.
- Funciona como base de datos en tiempo real.
- Contiene un sistema de seguridad por usuarios y también de modificación de nodos por reglas.

Conclusión

Se puede decir que todas las teorías sobre las tecnologías de las cuales se hicieron mención anteriormente fueron necesarios para realizar análisis comparativos sobre que tecnologías serian mas apropiadas para la aplicación móvil a desarrollar, es importante mencionar que aunque la aplicación a desarrollar se hará como una aplicación nativa del sistema operativo Android, también requerirá de programación web (Javascript PHP) para desarrollar un sistema que va a monitorear las opciones que contiene la base de datos, esto como requisito del usuario final. Para que se pueda comprender

totalmente en su totalidad basados en el tipo de aplicación y el sistema operativo seleccionado se decide que Java es el lenguaje de programación apto para programar, utilizando como herramienta de ayuda el IDE Android Studio y teniendo una base de datos del proveedor Google la cual es Firebase Database, una base de datos NoSQL con motivo de aumentar la cantidad de información que puede ser almacenada sin que haya necesidad de hacer un sacrificio de rendimiento en el dispositivo móvil pues este tipo de base de datos esta diseñada para soportar una gran cantidad de almacenamiento.

3 Estado del arte

Realizar una recopilación de conocimientos puede ayudar a conformar la aplicación basándose en productos o aplicaciones que ya existen similares al proyecto que se va a llevar a cabo, por lo que puede considerarse un marco referencial o Estado del arte, ya que se recopilan conocimientos de un área específica y así sistematizar esa información para tomar lo útil. Esto con objetivo de desarrollar una aplicación integrando características o funcionalidades que ya existen e innovando en otras diferentes. En este marco referencial se estudiarán algunas aplicaciones que fueron programadas para instituciones y que contienen funciones que son llamativas, sin embargo, la medida de impacto de la aplicación que se pretende desarrollar debe de ser diferente puesto que aunque se tomara guía de algunas aplicaciones ya existentes también va a contener un grado de innovación. La mayoría de las aplicaciones que aquí se hará mención son extraídas de la Play Store, ya que sus desarrolladores han logrado publicar una prueba piloto de su aplicación o ya esta totalmente liberada y estando en la Play Store cualquier usuario tiene acceso a estas.

3.0.1. Instituto Cumbres Mérida

Descripción de la Play Store

El instituto Cumbres Mérida es parte de una red internacional de colegios con más de 60 años de experiencia implementando el modelo de formación integral de los Legionarios de Cristo en 18 países del mundo.

Queremos formar personas y familias íntegras que promuevan y defiendan los principios del humanismo Cristiano.

Brindamos el apoyo necesario para el desarrollo de familias católicas que sepan responder a los retos de nuestros tiempos.

En estos tiempos de constantes cambios tecnológicos les brindamos a nuestros alumnos y padres de familia una nueva herramienta de comunicación desde la cual podrán tener acceso a :

- Circulares
- Agendar citas
- Calendarios
- Avisos escolares

Examinación por parte del Investigador

La aplicación Instituto Cumbres Merida es una app para el uso de la institución como un punto publicitario que contiene muchas funciones para establecer contacto con la Institución y además estar informado sobre avisos importantes que se publican en la misma aplicación. En resumen las funciones examinadas son las siguientes:

- Notificaciones. A través de una actividad de la aplicación hacen llegar las noticias más importantes que se publican dentro de la misma aplicación.
- Calendario. Posee un calendario de fechas importantes para los diferentes niveles que oferta la Institución que son: Preescolar, Primaria, Secundaria y Bachillerato. Este calendario funciona con un visor de documentos que visualiza un PDF que contiene un calendario con las fechas de cada una de las opciones que se pueden elegir.
- Vídeos. Cuenta también con un visor de vídeos sobre los eventos y publicaciones en vídeo importantes sobre la Institución.
- Circulares. Además de una opción para subir documentos importantes sobre la escuela, documentos como horarios, circulares, reglamentos, resultados de calificaciones y documentos de procesos educativos.
- Novedades. Dentro de este apartado del menú incluyen la opción de listar todos los proyectos de interés del estudiantado culminados o en proceso.
- Galería. En esta opción se lleva al usuario a una actividad con una lista de álbumes con fotos de la institución, dentro de cada álbum se despliega un visor personalizado y una opción para iniciar una presentación con todas las imágenes del álbum.

- Social Media. A través de la conexión de inicio de sesión con Facebook o Twitter, el usuario puede compartir el contenido de la aplicación en sus redes sociales sin necesidad de dejar la aplicación.
- Sugerencias / Comentarios y Contactanos. Esta opción muestra una actividad con un formulario que puede llenar un usuario con algún comentario sobre la aplicación además de proporcionar sus datos de contacto.
- Ubicación. Muestra una actividad que contiene un visor de Google Maps con todas las opciones de navegación inhabilitadas.
- Nuestro Sitio. Es una opción que permite visualizar la página del Instituto Cumbres Merida a través de un web view.

3.0.2. MiTec Móvil

Descripción de la Play Store

Con el fin de continuar fortaleciendo la integración de la tecnología móvil en los servicios académicos y administrativos que el Tecnológico de Monterrey ofrece a sus alumnos y mantener así una comunicación constante a través de los medios que les son cada vez más cercanos, los alumnos ya pueden descargar de forma gratuita a sus dispositivos móviles la aplicación “MiTec Móvil” (Apple, Android y BlackBerry) para consultar algunos de los servicios que se ofrecen actualmente a través del Portal de Alumnos MiTec.[32]

MiTec Móvil te ayudará a mantenerte informado con las últimas noticias del campus, ver vídeos relacionados a tu vida estudiantil, revisar tus calificaciones, navegar por tu catálogo de cursos en línea. También podrás configurar que servicios quieres ver en tu menú. Y todo esto en la palma de tu mano.

En esta nueva versión la aplicación contará con los siguientes servicios[32]:

- Boleta de calificaciones: Revisa tus calificaciones y faltas cuando estén disponibles; podrás consultar también las de semestres anteriores.
- Horario: Revisa tu horario de clases para el semestre
- Historia Académica: Revisa todas tus materias cursadas durante tu vida en el ITESM

- En Contacto: Tienes alguna sugerencia comentario o problema con algún servicio o instalaciones de tu campus, contáctanos para ayudarte
- Blackboard learn: Accede a tus cursos en línea. Videos: Ve los vídeos institucionales y los relacionados a eventos en los campus.
- Noticias: ¡Extra! Extra! Manténte al día con todo lo que sucede dentro de los campus.
- Asistencia (soporte): ¿Tienes dudas o preguntas sobre MiTec Móvil? Escríbenos
- Biblioteca: Encuentra recursos en la biblioteca. Podrás ver la disponibilidad actual y solicitar libros.
- iTunes U: Escuchar o descarga nuestra colección de grabaciones de audio y vídeo, incluyendo cursos y conferencias.
- Mapas: Localiza los Campus del país y traza la ruta de navegación desde tu ubicación actual.
- Eventos de Exatec y CVC: Mantente informado de los eventos de tu interés y añádelos a tu calendario.
- Bolsa de Trabajo: Busca dentro de nuestras vacantes disponibles según tus preferencias y agrega a tus favoritos los trabajos que desees revisar.

Examinación por parte del Investigador

La aplicación MiTec Móvil es desarrollada por alumnado del Tecnológico de Monterrey cuyas funcionalidades están oriendas al estudiantado y solo algunas disponibles para promoción de la oferta academia y comunicación con la institución a través de información de contacto. Proporciona información sobre sus diversos campus y sus ubicaciones ademas de promocionar la institución a través de diferentes opciones. Existen funciones que están reservadas para los estudiantes y solo se pueden acceder a través de un usuario y contraseña, por ello solo se hará mención de las funciones disponibles al publico en general.

- Oferta Educativa. Nos muestra una actividad que permite seleccionar dos opciones, seleccionar área o campus, están divididas de esta manera para agrupar la oferta académica, al seleccionar área, nos aparece un listado de todas las áreas que existen en las carreras ofertadas por la institución, al seleccionar campus,

nos muestra una lista de todos los campus de la institución, y las áreas que tiene cada uno.

- TecMovil. Es una opción que lleva a una actividad que contiene todas las aplicaciones que el Tecnológico de Monterrey ha lanzado en la Play Store.
- ITUNES U. A través de la aplicación iTunes, la aplicación ofrece la colección de música y vídeos de diferentes eventos recopilados para disposición del usuario de la aplicación y conocer actividades de la institución.
- Campus. Ofrece un listado de todos los campus, al seleccionar alguno de la lista muestra un mapa con la ubicación de este.
- Videos. Es una opción con un link directo al canal de YouTube del Tecnológico de Monterrey.

3.0.3. Guía UNAM

Descripción en la Play Store

¿Tu examen de ingreso a la UNAM se avecina? No te preocupes, ponte cómodo, abre tu Guía UNAM. Haz un examen simulación para evaluar tus conocimientos. Estudia los temas que fallan y toma nota.

Aprende a tu ritmo y estudia como nunca. Todos los temas están desarrollados por materia, organizados y redactados en un lenguaje claro para que solo te enfoques en estudiar.

Estudia todas las materias que vendrán en tu examen de admisión; Matemáticas, Física, Química, Biología, Geografía, Historia Universal, Español, Historia de México, Literatura y Filosofía.[28]

Examinación por parte del Investigador

La aplicación esta realizada para la practica del Test de Ingreso a la UNAM, para ello requiere de utilizar algunos datos sobre las redes sociales del usuario para ofrecerle algunas funcionalidades y también como recopilación de resultados como información útil para la institución. Las funciones con las que cuenta la app son para que el usuario pueda tener acceso a recursos relevantes sobre los exámenes de admisión.

- Estudia. Permite seleccionar alguna de las múltiples lecciones que se califican en el Examen teniendo agrupados los temas.
- Notas. Dentro de la actividad de ESTUDIA se muestran iconos que permiten almacenar notas o recordatorios, estos se guardan y muestra en la actividad de Notas.
- Examen de materias. Permite que se realicen algunos de los exámenes básicos de la institución con motivo de practica.
- Examen simulación. Simula un examen con todas las áreas a calificar.

Conclusión

En este capítulo se incluyeron algunas de las aplicaciones que ya están en uso y contienen muchas de funciones útiles para los aspirantes a ingresar a sus instituciones. En todos los casos estudiados se puede notar que las aplicaciones son desarrolladas por instituciones de alto impacto, por lo que se considera importante realizar proyectos como este que resalten el uso de las nuevas tecnologías en alguna institución, esto la hace llamativa para el aspirante ya que a cierto modo la aplicación concede al usuario estar en contacto con la institución. Por ello se considera importante que todo lo que se incluye en la aplicación móvil que se desarrolla en este proyecto debe ser de interés y de utilidad para el usuario.

4 Marco metodológico

Para este trabajo de tesis es necesario incluir una metodología o marco metodológico que defina la disciplina que elabora y sistematiza el conjunto de procedimientos para elaborar la investigación, de esta manera seleccionar las técnicas para la búsqueda de datos y la construcción del conocimiento de la investigación. El marco metodológico consiste entonces en un conjunto más o menos coherente y racional de técnicas y procedimientos cuyo propósito fundamental apunta a implementar procesos de recolección, clasificación y validación de datos y experiencias provenientes de la realidad, y a partir de los cuales pueda construirse el conocimiento para llevar a cabo la investigación y a la obtención de sus resultados.

4.1. Enfoque de la investigación

A lo largo de la Historia de la Ciencia han surgido diversas corrientes de pensamiento (como el empirismo, el materialismo dialéctico, el positivismo, la fenomenología, el estructuralismo) y diversos marcos interpretativos, como la etnografía y el constructivismo, que han originado diferentes rutas en la búsqueda del conocimiento. Sin embargo, y debido a las diferentes premisas que las sustentan, desde el siglo pasado tales corrientes se han “polarizado” en dos aproximaciones principales para indagar: el enfoque cuantitativo y el enfoque cualitativo de la investigación. Ambos enfoques emplean procesos cuidadosos, metódicos y empíricos en su esfuerzo para generar conocimiento, por lo que la definición previa de investigación se aplica a los dos por igual, y utilizan, en términos generales, cinco fases similares y relacionadas entre sí[35]:

1. Llevan a cabo la observación y evaluación de fenómenos.
2. Establecen suposiciones o ideas como consecuencia de la observación y evaluación realizadas.
3. Demuestran el grado en que las suposiciones o ideas tienen fundamento.

4. Revisan tales suposiciones o ideas sobre la base de las pruebas o del análisis.
5. Proponen nuevas observaciones y evaluaciones para esclarecer, modificar y fundamentar las suposiciones e ideas; o incluso para generar otras.

El enfoque cuantitativo (que representa, como dijimos, un conjunto de procesos) es secuencial y probatorio. Cada etapa precede a la siguiente y no podemos “brincar o eludir” pasos, 3 el orden es riguroso, aunque, desde luego, podemos redefinir alguna fase. Parte de una idea, que va acotándose y, una vez delimitada, se derivan objetivos y preguntas de investigación, se revisa la literatura y se construye un marco o una perspectiva teórica. De las preguntas se establecen hipótesis y determinan variables; se desarrolla un plan para probarlas (diseño); se miden las variables en un determinado contexto; se analizan las mediciones obtenidas (con frecuencia utilizando métodos estadísticos), y se establece una serie de conclusiones respecto de la(s) hipótesis.

El enfoque cualitativo también se guía por áreas o temas significativos de investigación. Sin embargo, en lugar de que la claridad sobre las preguntas de investigación e hipótesis preceda a la recolección y el análisis de los datos (como en la mayoría de los estudios cuantitativos), los estudios cualitativos pueden desarrollar preguntas e hipótesis antes, durante o después de la recolección y el análisis de los datos. Con frecuencia, estas actividades sirven, primero, para descubrir cuáles son las preguntas de investigación más importantes, y después, para refinarlas y responderlas. La acción indagatoria se mueve de manera dinámica en ambos sentidos: entre los hechos y su interpretación, y resulta un proceso más bien “circular” y no siempre la secuencia es la misma, varía de acuerdo con cada estudio en particular[35].

El enfoque empleado en esta investigación es el cualitativo ya que se va a requerir de datos que puedan medirse o contarse para la comprobación de la hipótesis.

4.2. Tipo de investigación

Para poder elegir un tipo de investigación primero se debe definir que es y que tipos existen, de esta manera se puede analizar este trabajo de tesis y ver que tipo aplica. Los tipos de investigación son todas aquellas diferentes maneras de llevar a cabo un estudio sobre cierto tema, estos tipos de investigación se dividen en 4 grupos que dependen de diferentes objetivos, métodos de estudio o procedimientos. A continuación se describen las 4 agrupaciones y los diferentes tipos de investigación[35]:

4.2.1. Según el propósito u objetivo de la investigación

1. Investigación pura o teórica. Este tipo de investigación tiene como principal objetivo la obtención de conocimientos de diferente índole, sin tener en cuenta la aplicabilidad de los conocimientos obtenidos. Gracias al cuerpo de conocimientos extraídos de ella pueden establecerse otro tipo de investigaciones, o no.
2. Investigación aplicada. Se trata de un tipo de investigación centrada en encontrar mecanismos o estrategias que permitan lograr un objetivo concreto, como curar una enfermedad o conseguir un elemento o bien que pueda ser de utilidad. Por consiguiente, el tipo de ámbito al que se aplica es muy específico y bien delimitado, ya que no se trata de explicar una amplia variedad de situaciones, sino que más bien se intenta abordar un problema específico.

4.2.2. Nivel de profundización del conocimiento

1. Investigación exploratoria. Este tipo de investigación se centra en analizar e investigar aspectos concretos de la realidad que aún no han sido analizados en profundidad. Básicamente se trata de una exploración o primer acercamiento que permite que investigaciones posteriores puedan dirigirse a un análisis de la temática tratada. Por sus características, este tipo de investigación no parte de teorías muy detalladas, sino que trata de encontrar patrones significativos en los datos que deben ser analizados para, a partir de estos resultados, crear las primeras explicaciones completas sobre lo que ocurre.
2. Descriptiva. El objetivo de este tipo de investigación es únicamente establecer una descripción lo más completa posible de un fenómeno, situación o elemento concreto, sin buscar ni causas ni consecuencias de éste. Mide las características y observa la configuración y los procesos que componen los fenómenos, sin pararse a valorarlos. Así pues, en muchas ocasiones este tipo de investigación ni siquiera se pregunta por la causalidad de los fenómenos (es decir, por el "por qué ocurre lo que se observa"). Simplemente, se trata de obtener una imagen esclarecedora del estado de la situación.
3. Explicativa. Se trata de uno de los tipos de investigación más frecuentes y en los que la ciencia se centra. Es el tipo de investigación que se utiliza con el fin de intentar determinar las causas y consecuencias de un fenómeno concreto. Se busca

no solo el qué sino el porqué de las cosas, y cómo han llegado al estado en cuestión. Para ello pueden usarse diferentes métodos, como la el método observacional, correlacional o experimental. El objetivo es crear modelos explicativos en el que puedan observarse secuencias de causa-efecto, si bien estas no tienen por qué ser lineales (normalmente, son mecanismos de causalidad muy complejos, con muchas variables en juego).

4.2.3. Según los tipos de datos a trabajar

1. Investigación cualitativa. Se trata de una investigación que nos lleva a la obtención de datos que nos son cuantificables, de manera que obtenemos una gran cantidad de información, pero hay subjetividad y una falta importante de justificación de los fenómenos. Sin embargo suele ser una fase inicial que nos permite posteriormente obtener una información más completa y objetiva.
2. Investigación cuantitativa. Este es el paso en el que se lleva a cabo un análisis y estudio de la realidad, es decir, se trata de una investigación objetiva en la que se establecen mediciones reales con lo que se obtiene una mayor cantidad de datos fiables, buscando así explicaciones contrastadas, estadísticas y generalizables.

4.2.4. Según el grado de manipulación de las variables

1. Investigación experimental. Este tipo de investigación se basa en la manipulación de variables en condiciones altamente controladas, replicando un fenómeno concreto y observando el grado en que la o las variables implicadas y manipuladas producen un efecto determinado. Los datos se obtienen de muestras aleatorizadas, de manera que se presupone que la muestra de la cual se obtienen es representativa de la realidad. Permite establecer diferentes hipótesis y contrastarlas a través de un método científico.
2. Cuasi-experimental. La investigación cuasi-experimental se asemeja a la experimental en el hecho de que se pretende manipular una o varias variables concretas, con la diferencia de que no se posee un control total sobre todas las variables, como por ejemplo aspectos vinculados al tipo de muestra que se presenta al experimento.

3. No experimental. Este tipo de investigación se basa fundamentalmente en la observación. En ella las diferentes variables que forman parte de una situación o suceso determinados no son controladas.

Para llevar a cabo una investigación no existe una norma de que diga que solo se debe elegir un tipo de investigación para reflejar la esencia de la misma. Por ello los tipos de investigación que se emplean en este trabajo de tesis son 2:

- Cuasi-experimental: Este tipo de investigación es empleado por que se van a analizar los comportamientos de las variables al sufrir ciertos cambios provocados por el investigador que también es importante señalar no tendrá el control total de todas las variables planeadas en la hipótesis por lo que no podrán ser alteradas o controladas al 100 %. Un ejemplo que es que no se pueden manipular las aptitudes y actitudes de los aspirantes que serán participantes en una simulación para medir su desempeño académico después de utilizar la aplicación. Mas sin embargo un ejemplo de una variable que si puede sufrir control es la implementación de la aplicación por lo que se podrán medir variables como las preferencias de los estudiantes y el alcance geográfico.
- Cuantitativa: Este tipo de investigación se emplea debido a que es mas confiable un resultado que es respaldado con datos seguros o contabilizados. Llevando a cabo este tipo de investigación, se tiene un conocimiento exacto de los resultados y el por que de los mismos, evaluar en que puntos de la investigación puede haber mejoras y en cuales hubo resultados aceptables o satisfactorios. Todo esto por que se utilizaran herramientas que permitan medir de manera real los resultados que se vayan obteniendo durante la investigación.

4.3. Alcance de la investigación

Ahora que se ha definido que la investigación será cuantitativa es necesario definir el alcance que tendrá esta. El libro de metodología de la investigación de Sampieri, Fernández y Batista define 4 alcances de que puede tener el proceso de investigación cuantitativa[35].

1. Alcance exploratorio: El alcance de los estudios exploratorios se realizan cuando el objetivo es examinar un tema o problema de investigación poco estudiado,

del cual se tienen muchas dudas o no se ha abordado antes. Es decir, cuando la revisión de la literatura reveló que tan sólo hay guías no investigadas e ideas vagamente relacionadas con el problema de estudio, o bien, si deseamos indagar sobre temas y áreas desde nuevas perspectivas.

2. Alcance descriptivo: El alcance de los estudios descriptivos buscan especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis. Es decir, únicamente pretenden medir o recoger información de manera independiente o conjunta sobre los conceptos o las variables a las que se refieren, esto es, su objetivo no es indicar cómo se relacionan éstas.
3. Alcance correlacional: Este tipo de estudios tiene como finalidad conocer la relación o grado de asociación que exista entre dos o más conceptos, categorías o variables en un contexto en particular. En ocasiones sólo se analiza la relación entre dos variables, pero con frecuencia se ubican en el estudio relaciones entre tres, cuatro o más variables. Los estudios correlacionales, al evaluar el grado de asociación entre dos o más variables, miden cada una de ellas (presuntamente relacionadas) y, después, cuantifican y analizan la vinculación.
4. Alcance explicativo: El alcance de los estudios explicativos van más allá de la descripción de conceptos o fenómenos o del establecimiento de relaciones entre conceptos; es decir, están dirigidos a responder por las causas de los eventos y fenómenos físicos o sociales. Como su nombre lo indica, su interés se centra en explicar por qué ocurre un fenómeno y en qué condiciones se manifiesta, o por qué se relacionan dos o más variables.

En este trabajo de investigación esta utilizándose el alcance correlacional, esto debido a que en las hipótesis establecidas contienen variables dependientes e independientes y se evalúan la relación que existen entre estas. La relación que se analiza en la primer hipótesis es; la implementación de una aplicación móvil (variable independiente) dará un mayor alcance geográfico (variable dependiente); en la segunda hipótesis se analiza la relación de el desempeño académico (variable dependiente) y la orientación de acuerdo a las aptitudes y actitudes del aspirante (variable independiente); en la tercer hipótesis se analiza la preferencia de los estudiantes en menor tiempo (variable dependiente) de acuerdo a las funciones que se ofertan en el ITSMT (variable independiente).

4.4. Diseño de la investigación

Para el diseño de la investigación de este trabajo de tesis se ha elegido el experimental, puesto que se van a comprobar los efectos que tendrá la intervención de una aplicación en el proceso de promoción y difusión del ITSMT, para realizar la comprobación el investigador deberá estar activo y deberá ser participante de los sucesos que se mencionaron anteriormente en la descripción del problema, ya que la aplicación genera un impacto positivo o negativo este deberá ser registrado, deberá haber recolección de datos y realizar un análisis de los mismos para comprobar las hipótesis[35].

4.5. Cronograma de actividades

Para mostrar de manera gráfica las actividades que se realizaron durante la investigación en la Figura 4.1 se muestra un cronograma que define las actividades y en que tiempo se realizaron de acuerdo al proceso de desarrollo de la tesis. Cabe mencionar que el proceso de investigación se realizo en tiempo y forma sin presentar alguna problemática u obstáculo que obstruyera el desarrollo de esta.

TIEMPOS TENTATIVOS		AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE				DICIEMBRE				ENERO				FEBRERO				MARZO			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
ACTIVIDADES	Generación de idea y elección del tema	x	x																														
	Problemática			x	x																												
	Desarrollo de objetivos					x	x																										
	Antecedentes y estado del arte							x																									
	Generación de hipótesis y definición de variables							x	x																								
	Revisión de literatura y desarrollo de marco teórico									x	x																						
	Desarrollo del diseño de investigación											x	x	x	x																		
	Definición de unidades de análisis															x																	
	Elaboración de herramientas de recolección de datos															x	x	x															
	Recolección de datos																			x	x	x											
	Análisis de los resultados																					x											
	Elaboración de reporte de resultados																								x	x	x	x					

Figura 4.1: Cronograma de actividades. Fuente: Elaboración propia.

4.6. Unidades de análisis y técnicas de recolección de datos.

Para llevar a cabo el estudio e impacto de la aplicación como una herramienta de promoción educativa es necesario observar su desempeño con un grupo de usuarios, dicha actividad en lugar de medir el funcionamiento de la aplicación calificaría la

comodidad y experiencia que el usuario tiene con la aplicación, ya que su opinión define si las hipótesis planteadas anteriormente se cumplen o no. Recopilar estos datos es indispensable para seguir mejorando la aplicación y también adaptar muchos de los procesos de promoción y difusión a las tecnologías móviles de hoy en día. Por ello esta recopilación de datos además de calificar a la aplicación pueden ser utilizados para mejorar el proceso de promoción y difusión de una institución educativa.

4.6.1. Unidades de análisis

Universo

Dentro de las unidades de análisis se ha establecido que todo el universo serán las localidades de la región cercana a la ciudad de Martínez de la Torre Veracruz ya que todas ellas podrán o no interactuar directa o indirectamente con el ITSMT.

Población

La población de la investigación serán los usuarios de la aplicación. De toda esta población se puede tomar una muestra, la cual será una cantidad finita de usuarios que ya hayan interactuado o que estén en curso de interactuar con la app.

Muestra

El procedimiento de obtención de la muestra será de forma aleatoria de esta forma todo cliente de la población tiene posibilidad y no hay manipulación alguna en esta selección. Para la selección se solicitó el apoyo de una institución educativa de bachillerato que puso a disposición del investigador un grupo de 30 estudiantes de 6to semestre de diferentes áreas para utilizar la aplicación y calificar el desempeño e impacto en el proceso de publicidad.

4.6.2. Técnicas de recolección de datos

Las técnicas de recolección de datos que se utilizaran en esta investigación son la encuesta y la observación. La encuesta que se va a realizar es conforme a la calificación que los participantes de la actividad de recolección de datos le den a la aplicación de acuerdo al uso que tuvieron con esta. La observación sera por parte del investigador, en su actividad como observador deberá responder un cuestionario sobre como califica la habilidad de un usuario para comprender la aplicación y ver si resulta fácil de utilizar o si los procesos que contiene son muy complicados. Cambien va a definir si los usuarios se sienten apoyados por la aplicación para poder escoger una carrera que oferte el ITSMT.

El objetivo de la experimentación consiste en lograr la identificación de los patrones secuenciales de los usuarios dentro de la plataforma, así como la obtención del conjunto de datos candidatos, es decir aquellos datos que me resultaran indispensables para llegar a realizar una predicción y por ultimo le creación de reglas de asociación estas con la intención de poder predecir el futuro comportamiento de los usuarios en base a la utilidad que los usuarios le da a determinadas actividades dentro de la plataforma.

5 Desarrollo e Implementación de la app

La app se va a desarrollar utilizando una metodología propuesta por el investigador, ya que al no haber ninguna metodología aceptada como estándar único para desarrollo de software móvil se pretende que la metodología que se utilizara en este proyecto sea también una propuesta para el desarrollo de aplicaciones móviles. Dentro de esta metodología se abordarán los siguientes pasos: Etapa 1 recolección y análisis de requerimientos, donde se van a recopilar todos los requisitos que debe de cumplir la app y se hará un análisis sobre que tecnologías pueden solventar esas necesidades. Etapa 2 diseño rápido, esta etapa nos va a permitir pre diseñar y después diseñar en base a las ideas aceptadas sobre módulos que la aplicación pueda tener para que cubran las necesidades establecidas en el análisis acoplándola con una interfaz que cumpla con ser intuitiva, fácil de usar y agradable al usuario. Etapa 3 Planeación, después de realizar un diseño de trabajo acerca de como cubrir las necesidades hay que establecer una planeación que sirva de guía para saber como implementar los módulos diseñados anteriormente. Etapa 4 Modelado, en esta etapa se va a realizar un bosquejo de los módulos diseñados para ir adaptándolos al producto final con objetivo de que se afinen detalles antes de comenzar con el desarrollo de la aplicación. Etapa 5 Construcción y desarrollo, esta etapa es donde se realiza la escritura de código, diseño del producto y sus interfaces haciendo uso de tecnologías de programación que puedan satisfacer las necesidades requeridas por el cliente. Etapa 6 Pruebas y evaluaciones, a fin de obtener un producto de calidad se realizarán pruebas para detectar errores y/o incongruencias antes de su despliegue. Etapa 7 Liberación, donde finalmente la app se va a generar sin realizar más cambios considerándose totalmente finalizada.

5.1. Metodología de desarrollo de software a utilizar

Para poder llevar a cabo el desarrollo de una aplicación móvil, es necesario estructurar las actividades a realizar. Actualmente para el desarrollo de software móvil no existe

una metodología que ya se haya convertido en el estándar general. Por ello se analizaron distintas propuestas de metodologías y se obtuvieron las etapas que se creían útiles para el desarrollo de esta aplicación. Las metodologías de las que se absorbe conocimiento son las siguientes:

5.1.1. Metodología de desarrollo de software general de Pressman

La metodología general de desarrollo de cualquier tipo de software propuesta por Roger Pressman consta de 5 actividades en un proceso iterativo[18]. Esas fases son las siguientes:

1. Comunicación: Antes de que comience cualquier trabajo técnico, tiene importancia crítica comunicarse y colaborar con el cliente (y otros participantes). Se busca entender los objetivos de los participantes respecto del proyecto, y reunir los requerimientos que ayuden a definir las características y funciones del software.
2. Planeación: Cualquier viaje complicado se simplifica si existe un mapa. Un proyecto de software es un viaje difícil, y la actividad de planeación crea un “mapa” que guía al equipo mientras viaja. El mapa -llamado plan del proyecto de software- define el trabajo de ingeniería de software al describir las tareas técnicas por realizar, los riesgos probables, los recursos que se requieren, los productos del trabajo que se obtendrán y una programación de las actividades.
3. Modelado: Ya sea usted diseñador de paisaje, constructor de puentes, ingeniero aeronáutico, carpintero o arquitecto, a diario trabaja con modelos. Crea un “bosquejo” del objeto por hacer a fin de entender el panorama general -como se verá arquitectónicamente, como se ajustan entre si las partes constituyentes y muchas características más-, si se requiere, refina el bosquejo con más y más detalles en un esfuerzo por comprender mejor el problema y como resolverlo. Un ingeniero de software hace lo mismo al crear modelos a fin de entender mejor los requerimientos del software y el diseño que los satisfará.
4. Construcción: Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores en este.
5. Despliegue: El software (como entidad completa o como un incremento parcialmente terminado) se entrega al consumidor que lo evalúa y que le da retroalimentación, misma que se basa en dicha evaluación.

Esos son las actividades propuestas por Pressman para el desarrollo general de software por lo que puede ser aplicado también para el desarrollo de software móvil, de esta metodología se ha decidido extraer las etapas de planeación y modelado.

5.1.2. Propuesta de metodología de desarrollo de aplicaciones móviles a utilizar

En base al conocimiento que se toma de la metodología propuesta por Pressman se agregan las siguientes etapas de la metodología propuesta por Ponce Mendoza Ulises[17] quedando de la siguiente manera la metodología a utilizar:

1. Recolección y análisis de requerimientos. En esta etapa se enlistarán los requerimientos básicos y funciones principales que requiere el cliente para la aplicación.
2. Diseño rápido. En esta etapa se realiza un adelanto de los módulos que pueden cubrir los requisitos, teniendo una idea de lo que se busca realizar en cada modulo de acuerdo a lo solicitado.
3. Planeación. En esta etapa se realizará de acuerdo al diseño rápido y el análisis de requerimientos cuales serán los módulos que se aceptan y el orden de programación de los mismos.
4. Modelado. En esta etapa se van a modelar las interfaces que se van a requerir en base a la etapa del diseño rápido adaptándolas a los módulos para obtener los detalles que pueden interrumpir el proceso de desarrollo y así afinar el producto y sus funciones. También se va a diseñar la base de datos y la estructura que va a contener.
5. Construcción y desarrollo. En esta etapa se va a generar el código para que las interfaces puedan funcionar y cumplir sus objetivos.
6. Pruebas y Evaluaciones. En esta etapa se va a evaluar que los módulos funcionen correctamente de acuerdo a sus objetivos y puedan satisfacer también las necesidades del cliente.
7. Lanzamiento. Después de ser aprobada totalmente la aplicación previa a su lanzamiento se construye su manual de uso y se libera en totalidad la aplicación.

La metodología bajo la que se construye la aplicación móvil DIFUTEC es una metodología desarrollada en base a dos distintas por lo que se pueden considerar características positivas, aunque también podría tener algunas desventajas.

Características: La metodología al ser elegida en base a la de Pressman puede ser iterativa es decir puede cumplir con el software en proporciones pequeñas hasta su entrega total por lo que puede considerarse que esta metodología contiene la esencia de un proceso ágil, puesto que su desarrollo es incremental, otra característica es que esta metodología es cooperativa, ya que clientes y desarrolladores trabajan juntos constantemente para obtener el mejor producto de software, debe ser un método sencillo sus etapas no deben ser confundibles para que pueda ser fácil de aprender y modificar algunas actividades de este, y como última característica es que es adaptable ya que permite involucrarse a cualquier desarrollo de otro tipo de software de aplicación.

Para que el desarrollo de la aplicación pueda realizarse correctamente y además se explique de manera concreta, se hizo una comparación en el Cuadro 5.1 de los temas a abordar en este capítulo y que etapas de la metodología se abordarán en cada uno de ellos para así tener un correcto seguimiento del trabajo de investigación y el desarrollo de la aplicación.

Temas del capítulo	Etapas de la Metodología
Recolección y análisis de requerimientos	1. Recolección y análisis de Requerimientos
	2. Diseño rápido
Plan y guía de trabajo	3. Planeación
Diseño y creación de la base de datos	4. Modelado
Diseño de las pantallas de la app.	
Codificación Front-end y Back-end	5. Construcción y desarrollo
	6. Pruebas y evaluación
Implementación de la app desarrollada	7. Liberación

Cuadro 5.1: Tabla ilustrativa de los temas del capítulo en comparación con las etapas que se abordarán en cada uno. Fuente: Elaboración propia.

5.2. Recolección y análisis de requerimientos

Como primer paso en la construcción de cualquier software es necesario establecer una comunicación con el cliente para definir su producto, es importante aclarar el nivel de complejidad del mismo y muchos de los aspectos que conlleva un proyecto. Por el punto anterior a través de una entrevista es necesario recopilar las necesidades o requerimientos del cliente en el producto que desea que se desarrolle.

5.2.1. Etapa 1. Recolección y análisis de requerimientos.

En esta etapa se hará la recolección de requerimientos y principales funcionalidades que el cliente desea en la aplicación móvil.

Para poder determinar todos estos requisitos se lleva a cabo una investigación con el cliente y ver qué requisitos pueden ser cubiertos en la primera versión de esta app.

Los requerimientos se obtuvieron mediante técnicas de entrevista con el cliente para que en comunicación con él se pueda determinar los requisitos y el alcance de la aplicación, estos se obtuvieron mediante las preguntas que Pressman señala en su metodología[18], las cuales son:

1. ¿Qué es lo que se requiere?
2. ¿Qué procesos desea cubrir con software?
3. ¿Quiénes realizan dichos procesos o actividades?
4. ¿Hacia a quienes está dirigido su servicio?
5. ¿Cuál es el objetivo del producto que requiere?

De acuerdo a las preguntas anteriores se establecen los requisitos mediante las respuestas obtenidas del cliente.

1. Respuesta de pregunta 1. Se requiere un programa que pueda utilizarse en los celulares que tienen sistema Android ya que es el sistema operativo más utilizado actualmente en dispositivos móviles. Este programa debe ser usado para llegar a más jóvenes con la intención de que conozcan la institución. Además de ser el más utilizado, es importante mencionar que los dispositivos de la mayoría de los estudiantes utilizan Android por ser más accesible económicamente.
2. Respuesta de pregunta 2. Los procesos que se quieren cubrir son los de difusión de noticias importantes de la institución. Análisis estadístico de los aspirantes que solicitan información sobre LA institución, que carreras son más interesantes y de donde provienen los aspirantes al tecnológico. Se requiere poder ayudar a los aspirantes a elegir su carrera mediante un test de orientación de acuerdo a las carreras que se ofertan en la institución. Y conocer que propuestas de carreras futuras les agrada más a los aspirantes.
3. Respuesta de pregunta 3. Estos procesos son realizados por el personal del departamento de publicidad del ITSMT.

4. Respuesta de pregunta 4. Este servicio está dirigido a jóvenes de preparatoria o trabajadores de edades mayores de 25 años que desean cursar el nivel Superior.
5. Respuesta de pregunta 5. Es generar o causar un impacto positivo en la forma de difusión y publicidad que realiza el ITSMT, que el alcance sea mayor en la región y así se puedan obtener resultados y datos estadísticos que ayuden en la toma de decisiones tanto de aspirantes sobre qué carrera elegir, como de personal del ITSMT sobre carreras futuras a abrir.

Después de analizar las respuestas se obtiene una lista de requerimientos que se confirman con el cliente para verificar que sea lo que requiere en realidad y en el Cuadro 5.2 se muestra una separación entre requisitos funcionales y no funcionales.

- R01: La aplicación debe desarrollarse con un entorno y diseño específico para el ITSMT.
- R02: La aplicación debe poder visualizar gráficas estadísticas sobre los siguientes aspectos: de que preparatoria vienen más aspirantes, que carrera les interesa más a los aspirantes, de qué localidad de la región vienen más aspirantes, que propuesta de carrera les interesa más a los aspirantes, el género de alumnos que más utilizan la aplicación y el conteo total de alumnos por los mismos aspectos anteriormente mencionados.
- R03: La aplicación debe poder permitir registrarse a los usuarios con un correo electrónico y con una cuenta de Facebook.
- R04: La aplicación debe hacer una recolección de los siguientes datos personales: nombre, apellidos, escuela de procedencia, correo electrónico, número telefónico y género.
- R05: La aplicación debe hacer una recolección de las preferencias de los usuarios como lo son: primera opción de carrera a elegir, elección de las propuestas presentadas y de que institución provienen.
- R06: La aplicación debe mostrar la oferta académica que ofrece el ITSMT.
- R07: La aplicación debe mostrar los proyectos y las actividades extraescolares que tiene el ITSMT actualmente.
- R08: En la aplicación debe haber una opción que permita al usuario realizar un test de orientación educativa sobre las carreras que oferta el ITSMT actualmente.

- R09: La aplicación debe permitir mostrar las noticias más relevantes publicadas por fecha reciente en la página de Twitter del ITSMT.
- R10: La aplicación debe poder en un 80 % de los textos que contiene cambiar el idioma de acuerdo al lenguaje del dispositivo.
- R11: La aplicación debe contener un apartado para mostrar créditos de la aplicación, número de versión, datos relevantes de la institución y de contacto.
- R12: Se debe entregar un sistema web que sirva para monitorear las opciones de la base de datos y gráficas mencionadas anteriormente.

Requisitos funcionales	Requisitos no funcionales
R03, R04, R05, R06, R07, R08, R09, R11, R12.	R01, R02, R10.

Cuadro 5.2: Separación de requisitos funcionales y no funcionales. Fuente: Elaboración propia.

5.2.2. Etapa 2. Diseño rápido

Para el diseño rápido se efectuaron las actividades del diseño de los módulos a utilizar de la aplicación y un boceto rápido de las pantallas que interactuarían con dichos módulos. Pero antes que nada hay que agrupar los requisitos que se pretenden cubrir con cada módulo y de esta manera realizar un caso de uso de cada uno de estos que refleje la funcionalidad que se espera en cada actividad.

Los módulos de la aplicación que se analizaron y que pretenden cumplir con las necesidades del cliente están detallados en el Cuadro 5.3, se escribe una descripción y también los requisitos que va a cubrir el módulo propuesto:

Módulo	Descripción	Requisitos a cubrir
Registro de datos y preferencias	Este módulo tiene por objetivo registrar todos los datos personales del usuario y también las preferencias que se han solicitado en el apartado de requisitos, además de realizar el registro de nuevos usuarios ya sea con correo electrónico o con cuenta de Facebook.	R03, R04 Y R05.
Consultas estadísticas	Este módulo tiene por objetivo mostrar a los usuarios un reporte estadístico de las preferencias de otros usuarios de acuerdo a los puntos que se solicitaron, que carrera les interesa a mas aspirantes, de que escuela provienen, las localidades de donde provienen y el género.	R02
Notas informativas	Este módulo interpretará la información obtenida de la línea de usuario de publicaciones de Twitter de la cuenta del ITSMT para mostrarlas a través de un cliente y los usuarios puedan tener acceso a las notas informativas de la institución.	R09
Oferta académica	El módulo de oferta académica tomara la información de las carreras cargadas en la base de datos mostrando el nombre de la carrera, la descripción de esta y el perfil que ofrece.	R06
Proyectos y actividades extraescolares	Este módulo permite al usuario consultar información acerca de los proyectos y las actividades extraescolares actuales en el ITSMT.	R07
Test orientativo	En este módulo el usuario podrá realizar un test de orientación para arrojar un resultado sobre qué carrera de la oferta académica del ITSMT se recomienda para el usuario.	R08
Monitor	El módulo monitor será desarrollado en plataforma web para que dos tipos de usuario administrador y consultor puedan modificar y consultar las opciones de la base de datos de la aplicación.	R12

Cuadro 5.3: Tabla de módulos propuestos, descripción y requisitos que pretenden cubrir. Fuente: Elaboración propia.

5.3. Plan y guía de trabajo

Ahora que se ha analizado el proyecto en función de los requerimientos y de los resultados que se pretenden obtener es necesario generar la base o planificación que funcione como guía durante todo el trabajo de investigación y desarrollo del proyecto. En este tema se abordara a elaboración de múltiples diagramas que puedan ser de utilidad para el desarrollo del proyecto

5.3.1. Etapa 3. Planeación

Para la planificación del proyecto se utilizará el estándar del PMI[19] el cual brinda lineamientos, reglas y características para la dirección de proyectos. Este estándar dispone de un ciclo de vida de un proyecto, el cual está dividido en varias fases, en la cual se tomó como referencia la fase de planificación, en esta fase definen tareas, calendario o cronograma de actividades, estimación de costo en tiempo y dinero. Define además ciertos documentos que deben realizar para tener una buena planificación del proyecto de software.

Para la documentación de la planeación se mostraran los siguientes diagramas que ayudaran a la comprensión del funcionamiento de la aplicación móvil.

1. Diagrama de clases
2. Diagrama de casos de uso

5.3.1.1. Programa de trabajo de elaboración de proyecto (Cronograma de actividades)

El documento a presentar para la planeación es el cronograma de actividades el cual se presenta de forma resumida y en un formato detallado por actividad. La descripción de las actividades es presentada a continuación y tambien se muestra en la Figura 5.1 a detalle su orden de realización :

1. Análisis de necesidades:
 - ERS (Especificación de requerimientos de software). Se recopilan todos los requisitos y las necesidades que el usuario desea cubrir.

2. Análisis de módulos de la aplicación:

- Planeación de módulos a utilizar. Se realiza un análisis de las necesidades para determinar cómo pueden adaptarse a funcionalidades de algún módulo que solvente la necesidad.

3. Diseño de los módulos y pantallas:

- Planeación de la base de datos. Se realiza un boceto de la base de datos NoSQL que contendrá la información.
- Diseño de la base de datos. Se estructuran los nodos de la base de datos NoSQL que contendrá la información.
- Diseño de pantallas de módulos. Se van a diseñar las pantallas de la aplicación en Just In Mind[33].

4. Codificación de la aplicación por módulos:

- Conexión de la plataforma con la base de datos. Se realiza la codificación necesaria para conectar la aplicación y el sistema monitor a la base de datos NoSQL de Firebase[31].
- Codificación de los módulos de la aplicación. Codificación de los módulos de la aplicación y del sistema monitor web.

5. Prueba piloto:

- Obtención y generación de datos de prueba (Prueba Tester). Se interactúa con la aplicación y el sistema desde diversos dispositivos.
- Análisis de resultados. Se recopila información sobre la experiencia del usuario con la aplicación a través de encuestas.
- Evaluación de resultados. Se observa que el usuario este satisfecho con el uso de la aplicación.
- Corrección de errores y módulos con resultados no satisfactorios. Se corrigen los errores que se hayan presentado durante las pruebas.

6. Implementación total y liberación de la aplicación:

- Integrar todos los módulos de la aplicación. Sumar todos los módulos en la aplicación y el sistema monitor web.

- Liberación de la aplicación. Se libera el apk y el sistema monitor web.

7. Concluir documentación:

- Manual de usuario. Se genera un manual de usuario para la aplicación y el sistema monitor web.

Inicio:

07	08	2017
----	----	------

 Término:

10	12	2017
----	----	------

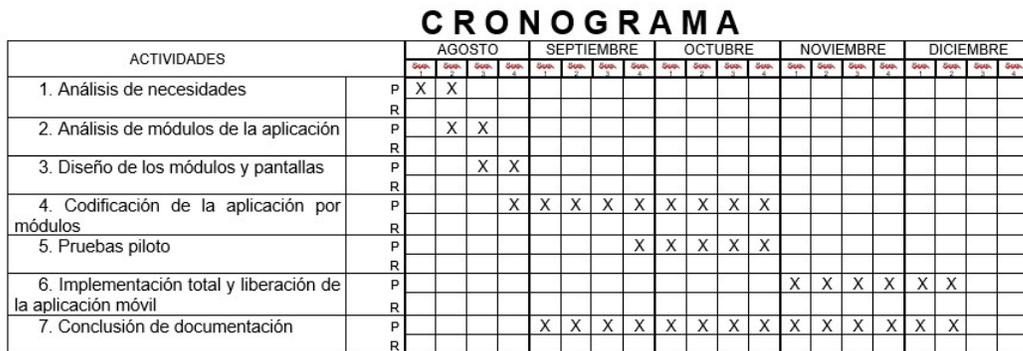


Figura 5.1: Cronograma de actividades de desarrollo del proyecto. Fuente: Elaboración propia.

5.3.1.2. Diagrama de clases

Para poder describir mejor un software es necesario crear una vista estética de las clases que lo van a componer para tener una idea clara de como funcionara la aplicación, en la Figura 5.2 se muestra el diagrama de clases de la aplicación y a continuación hay una descripción breve sobre cada entidad en el diagrama.

1. Aplicación móvil. Entidad padre que representa la app a generar, contiene 3 métodos y no requiere de ningún atributo.
2. Usuario. Entidad que representa al usuario con rol dentro de la app, dándole varios atributos que extienden los métodos básicos de la aplicación móvil.
3. Registrar_preferencias. Entidad o clase que representa al método que a su vez requiere de mas atributos del usuario que no puede proporcionar solo siendo usuario común. Cuenta con solo un método.
4. Consultar_estadísticas. Esta entidad no requiere de ningún atributo y el único método lo realiza en consulta a la base de datos.

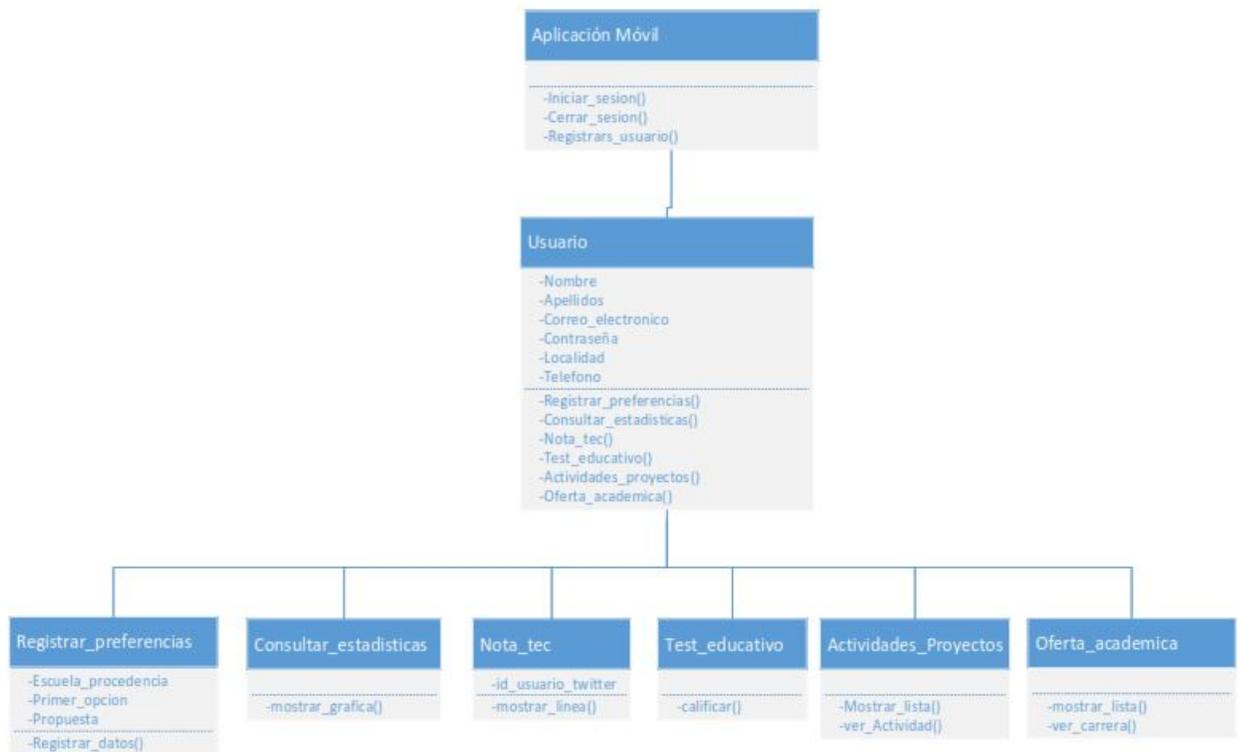


Figura 5.2: Diagrama de clases. Fuente: Elaboración propia.

5. Nota_tec. Contiene solo un atributo el cual es el id de usuario de la cuenta Twitter a enlazar, su método requiere forzosamente al atributo para poder funcionar.
6. Test_educativo. Una entidad muy interactiva pero sin atributos, ya que su método realiza una calificación de respuestas interactivas.
7. Actividades_proyectos y Oferta_académica: En estas entidades no se requieren ningún atributo ya que la información para interactuar y utilizar el método se solicitan en la base de datos.

5.3.1.3. Casos de uso

Una vez que se hizo mención de las entidades de la aplicación (basadas en los módulos de la aplicación) es necesario incluir algunos de los casos de uso que se elaboraron para

explicar y comprender el funcionamiento de la aplicación, de esta manera la percepción del usuario sobre la aplicación sera mas amplia. Un caso de uso es una unidad coherente de funcionalidad, expresada como transacción entre los actores y la aplicación. El propósito es enumerar a los actores (usuarios y entidades que interactúan con la aplicación) y los casos de uso y demostrar que actores participan en cada transacción[18].

A continuación en la Figura 5.3 y Figura 5.4 se mostraran dos casos de uso, uno de la aplicación móvil y otro del sistema monitor.

Caso de uso de la aplicación móvil - Registro de preferencias

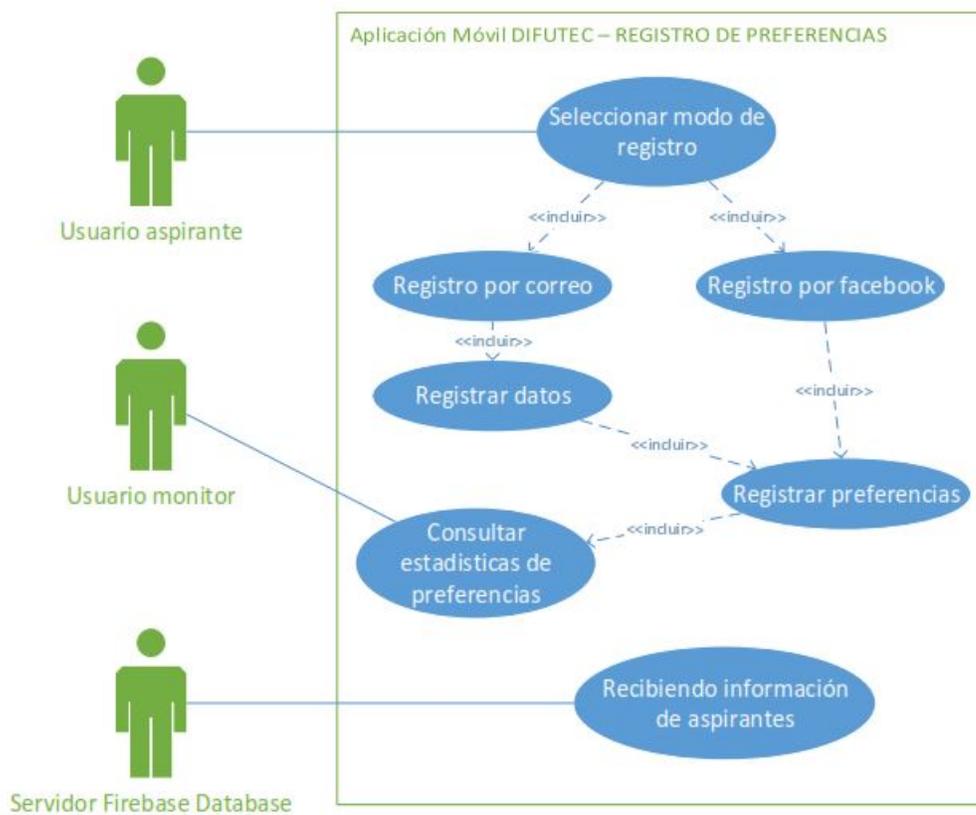


Figura 5.3: Caso de uso de la app - Registro de preferencias. Fuente: Elaboración propia

Descripción del caso de uso

Cuando un nuevo usuario instala la aplicación móvil en su dispositivo y la usa por primera vez, debe registrar sus datos y preferencias para recopilarlas y que puedan ser consultadas después en forma de gráfica por otros usuarios, para ello el usuario debe iniciar el proceso de registro y seleccionar la forma, si es por correo electrónico el usuario deberá llenar el formulario con una cuenta existente, o si el usuario selecciona el registro por Facebook, se extraerán los datos de la cuenta perfil activa actualmente en el móvil, una vez finalizado el registro de datos del usuario se envía a un formulario de registro de preferencias, donde el usuario responderá algunas preguntas de preferencias educativas. Al finalizar el registro, el usuario de la aplicación así como otro actor el cual puede ser un usuario monitor del sistema, pueden consultar las estadísticas de preferencias educativas por diversos rubros, mientras que otro actor, el cual es el servidor de la base de datos, estará recibiendo la información de los aspirantes o usuarios nuevos.

Caso de uso del sistema monitor - Administrador de usuarios**Descripción del caso de uso**

Cuando un usuario administrador entra al sistema monitor web, puede ingresar a una lista de usuarios o a al monitor para administrar la información de otros usuarios y propia también. En este caso de uso las conexiones entre algunos de los eventos, se dan por el conector “extender” el cual significa que ese evento se ofrece unicamente como opción, es decir no es un evento forzado, al ingresar un usuario administrador el cual es uno de los actores de este caso de uso, puede visualizar o una lista de usuarios o el monitor de estadísticas, en ambos puede realizar modificaciones de datos de usuarios, unicamente como opciones, ya sea o añadir un nuevo usuario o editar los datos de un usuario existente. Mientras que un usuario consultor solo puede visualizar al monitor para hacer consultas estadísticas y no visualiza una lista de administración de usuarios. Aunque a este actor también se le extienden 3 eventos con los cuales interactuá con su cuenta personal del sistema. Otro punto importante es señalar el conector entre el usuario administrador y el usuario consultor, estos dos usuarios al pertenecer al sistema web pueden ser similares en atributos o información, mas sus permisos de acceso son distintos, por ello el usuario consultor puede realizar un uso parcial (solo una parte) de las funciones que si puede realizar el usuario Administrador.

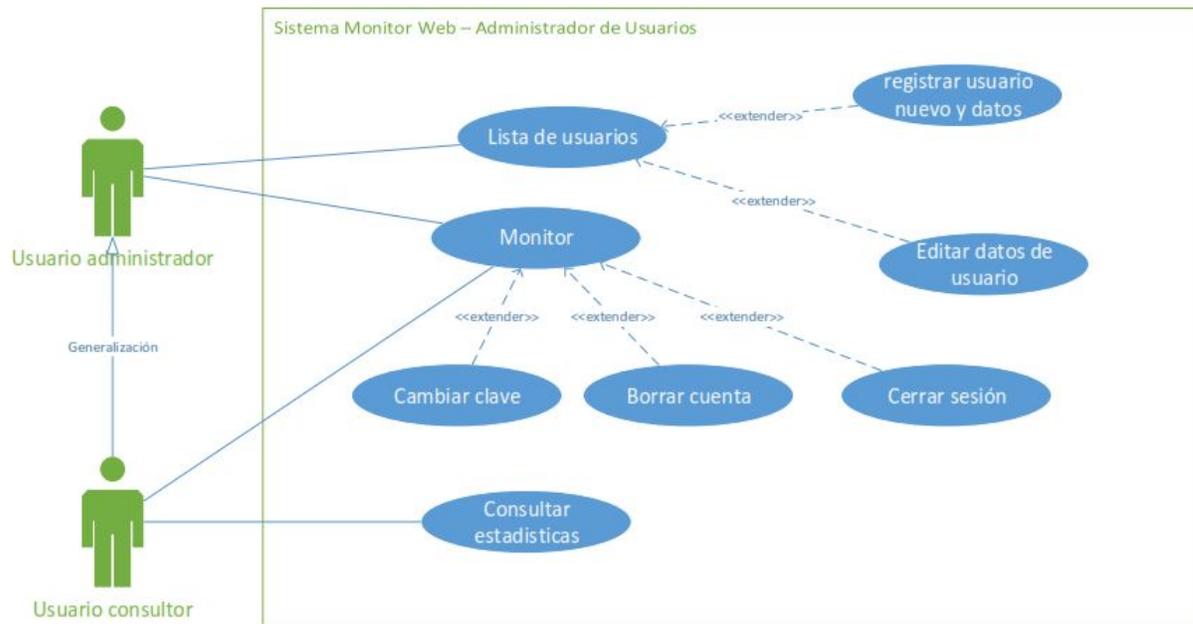


Figura 5.4: Caso de uso del monitor - Administrador de usuarios. Fuente: Elaboración propia.

5.4. Diseño de la base de datos

En los próximos 2 temas se abordara una sola etapa de la metodología la cual es muy importante por que se trata de la interfaz que se le va a dar a la aplicación ademas del diseño y creación de la base de datos, ambos temas son muy importantes puesto que una vez teniendo la planeación de desarrollo del proyecto, se debe proseguir con modelar el producto, es decir, como se vera la app al finalizar su construcción, o al menos un enfoque de como debe verse al final. Para poder llevar a cabo el diseño de los formularios y las pantallas también es necesario conocer los datos de entrada que tendrá cada una o las funciones con las que va a trabajar cada actividad (pantalla), esto es muy importante ya que en base a lo que queremos diseñar debemos estructurar nuestra base de datos para que pueda acoplarse a los formularios y trabajar en conjunto para tener un producto de calidad.

5.4.1. Etapa 4. Modelado

En esta etapa se lleva a cabo el modelado o diseño del producto de manera estética, es decir el como se vera al final del desarrollo, para poder interactuar con las pantallas es necesario primero diseñar y crear una base de datos que pueda contener la información que estos formularios o actividades van a enviar. Como todo producto de software, la base de datos debe diseñarse primero basándose en los módulos que se pretenden construir, los requerimientos solicitados por el cliente y por los objetivos de la aplicación.

5.4.1.1. Selección de la base de datos

Como primer punto debe seleccionarse un SGBD (Sistema Gestor de Bases de Datos) que pueda ser compatible con las características de la aplicación, en este caso, al ser una aplicación que pretende recopilar información de una gran cantidad de usuarios y que va a estar liberada al publico para que los usuarios puedan interactuar con ella, debe ser un SGBD que soporte el alto numero de peticiones y envíos de información.

Basados en las características principales que se requieren, escalabilidad y tolerancia a fallos. Se determina que una base de datos del tipo no relacional o NoSQL (no relacional) es la mejor opción para la aplicación que se pretende desarrollar. Pese a la no existencia de una definición formal, cuando hablamos de base datos NoSQL, también conocidas como “No sólo SQL”, nos referimos a una amplia clase de sistemas de gestión de datos (mecanismos para el almacenamiento y recuperación de datos) que difieren, en aspectos importantes, del modelo clásico de relaciones entre entidades (o tablas) existente en los sistemas de gestión bases de datos relacionales, siendo el más destacado el que no usan SQL como lenguaje principal de consulta.

Aunque son conocidas desde la década de los 60 del pasado siglo, su auge actual viene determinado por el uso que, de estos sistemas han hecho las principales compañías de Internet como Amazon, Google, Twitter y Facebook. Estas compañías tenían que enfrentarse a nuevos desafíos en el tratamiento de los datos motivados por el enorme crecimiento de la Web donde se requería dar respuesta a la necesidad de proporcionar información procesada a partir de grandes volúmenes de datos con unas estructuras horizontales, más o menos, similares y con aplicaciones web que debían dar respuesta a las peticiones de un número elevado e indeterminado de usuarios en el menor tiempo

posible. Estas compañías se dieron cuenta de que el rendimiento y sus necesidades de tiempo real eran más importantes que la consistencia de los datos, aspecto este último al que las bases de datos relacionales tradicionales dedicaban una gran cantidad de tiempo de proceso.

Al analizar los anteriores casos y considerar que el ITSMT no cuenta con un servidor en el que se soporten la gran cantidad de consultas en una base de datos, se considera que Firebase Database del fabricante GOOGLE INC es la mejor opción puesto que sus tecnologías están adaptadas a dispositivos móviles, ofreciendo también un servicio de almacenamiento de base de datos en los servidores de Google por lo que el ITSMT no requiere de un servidor para almacenar la base de datos.

5.4.1.2. Diagrama No Relacional

Actualmente no existe un diseño establecido para la elaboración de bases de datos no relacionales, por lo que un diagrama de entidad-relación no se utiliza para estos casos como lo suele ser en las bases de datos SQL. A diferencia de un diagrama entidad-relación lo que se pretende utilizar es un diagrama de grafos en forma de documento, los cuales representan a cada entidad de la base de datos, siendo algo parecido a las tablas de una base de datos relacional, pero en una base de datos no relacional ya no se conocen como tablas sino como nodos y los registros dejan de llamarse así para llamarse documentos que contienen la información derivada de cada nodo.

En la Figura 5.5 se muestra el diagrama no relacional de la base de datos de la aplicación móvil Difutec la cual también estará enlazada al monitor web por lo que esta base de datos podrá interactuar con ambos servicios.

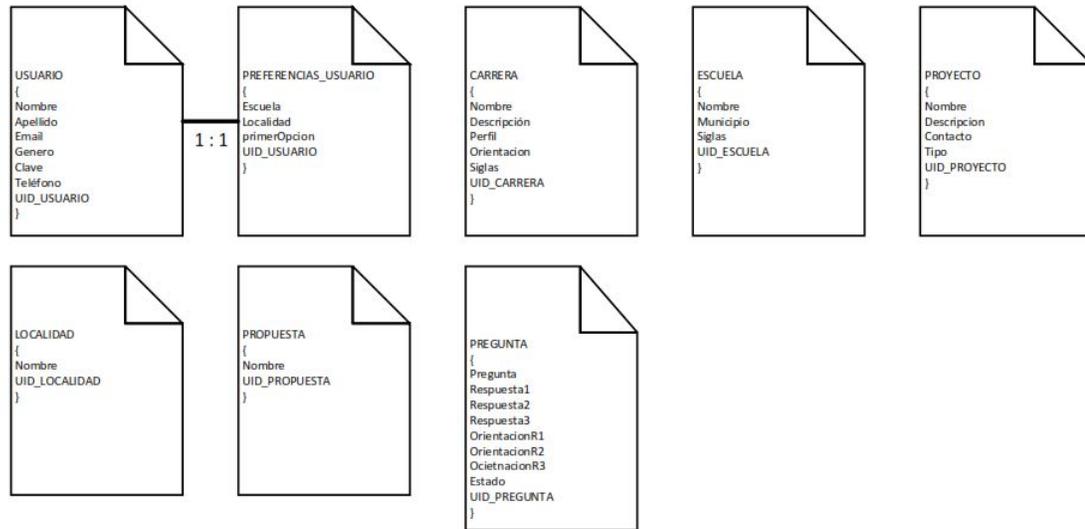


Figura 5.5: Diagrama por nodos de la app Difutec en forma de documentos. Fuente: Elaboración propia.

Como se aprecia en la imagen anterior, muchos de los nodos o documentos no contienen alguna relación entre ellos, a excepción de dos los cuales son Usuario y Preferencias_Usuario, estos dos documentos o nodos de la base de datos requieren que se estructure una relación, pero no quiere decir que se hará una llave foránea entre ambas, sino que igual que las bases de datos relacionales contendrán un nodo del documento que almacene el mismo dato para que al momento de realizar una consulta pueda verificarse si la información existe en ambos documentos, mas no quiere decir que hará dos consultas a dos tablas, si no que de manera estructurada se hará una consulta atómica que localice el campo UID_USUARIO el cual esta en ambos documentos, después se estructura un nodo JSON que contiene la información de la consulta y de esta manera llega al dispositivo que contiene un algoritmo que interpretas los nodos del formato JSON.

5.5. Diseño de las pantallas de la app

A continuación en este tema del capítulo se incluirán los diseños de las pantallas y se explicara acerca de la herramienta que se utilizo para llevarlos a cabo, como se mencionó en el tema anterior la etapa en la que se continua de la metodología es la etapa 4 de

modelado de la aplicación. Anteriormente se mostró un diagrama no relacional que contenía los documentos que se van a incluir en la base de datos, y se puede ver que se van a trabajar con formularios cortos pues la información que se enviara de la app a base de datos sera escasa ya que como requisito de funcionalidad un usuario no puede pasar todo el tiempo que esta en la aplicación escribiendo constantemente en sus formularios ya que provoca que sea poco atractiva, al contrario de un sistema normal en esta aplicación la poca información que recibe va a ser transformada en información visual de gran utilidad.

En base a lo anterior los diseños de las pantallas no deben estar llenos de formularios interminables que solo lleven al usuario al cansancio y desesperación por la gran cantidad de información que debe escribir. Por ello se requiere hacer un diseño de las pantallas antes de simplemente empezar a llenarlas de elementos que hagan que el usuario se confunda con la información que ingresa en estos. Para diseñar las pantallas se va a utilizar la herramienta Just In Mind la cual sirve para crear bocetos muy artísticos de pantallas de aplicación móvil, aplicaciones web y de escritorio. En este trabajo de investigación para no mostrar todos los diseños de las pantallas de la aplicación solo se incluirán 2 diseños y se explicara su función dentro de la aplicación[33].

Diseño de pantalla de menú principal

En la Figura 5.6 esta la pantalla donde se encuentra el menú principal, en su diseño podemos encontrar una lista de las opciones para entrar a cada uno de los módulos de la aplicación, esta lista puede ser deslizada y adaptarse al tamaño de la pantalla del dispositivo móvil para evitar ser redimensionada y perder calidad en imagen, dentro de la misma también se ubica un menú en la parte superior derecha el cual contiene las opciones principales del usuario que son para administrar su cuenta, en ellas puede editar su contraseña o cerrar sesión. en la parte superior del formulario se encuentra un saludo y un texto dinámico que se adapta al correo electrónico del usuario que esta utilizando la app en ese momento.

Diseño de pantalla de modulo de consultas estadísticas

En la Figura 5.7 hay una pantalla que va a mostrar un formulario interactivo con 3 elementos, en la parte superior izquierda de la pantalla hay un selector de opciones o

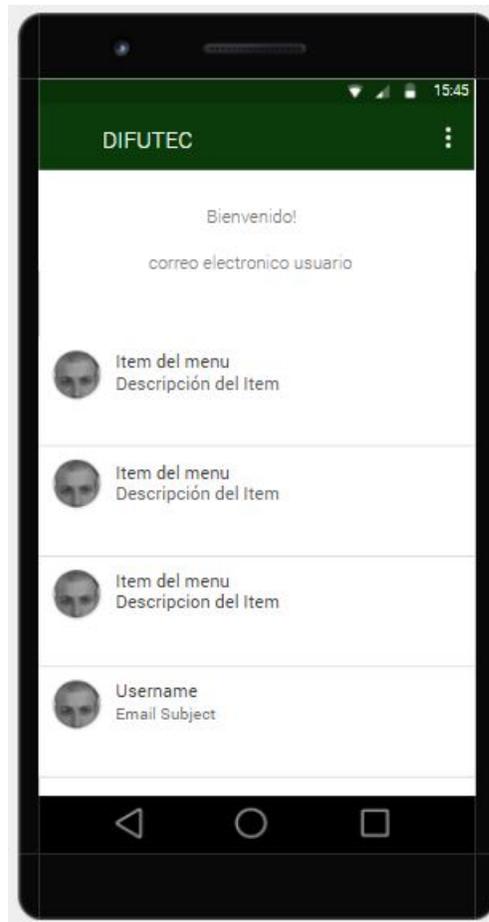


Figura 5.6: Diseño de pantalla del menú principal. Fuente: Elaboración propia.

Spinner como suele llamarse dentro del contexto de programación de Android Studio, este Spinner contiene las opciones por las cuales se permite mostrar la agrupación de los datos en forma de gráfica, estas opciones son por Escuela, Localidad, Genero, Propuesta y Carrera. En la parte superior derecha se encuentra un botón el cual ejecuta la acción de búsqueda después de seleccionar la opción para consultar. En la parte inferior se encuentra un lienzo donde se genera la gráfica con los resultados de la consulta en forma textual especificando el color y el nombre de la serie a la que pertenece para poderlos identificar en la gráfica.

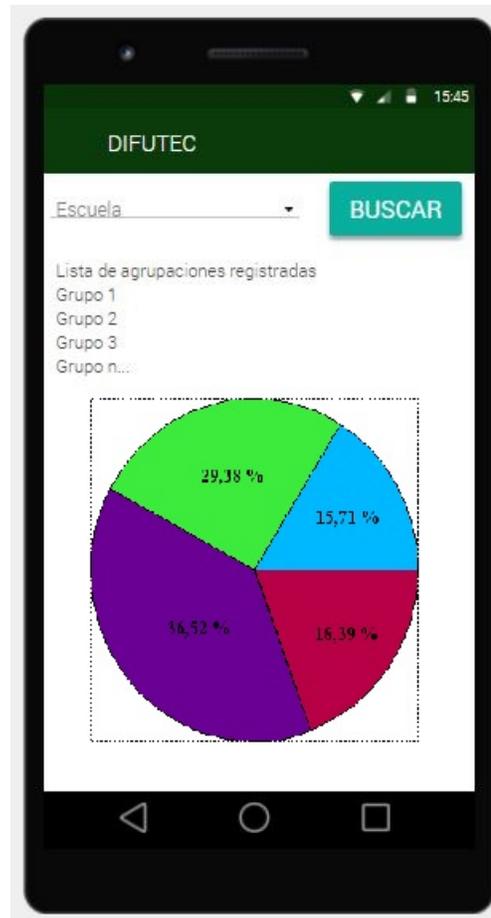


Figura 5.7: Diseño de pantalla del modulo de consultas estadísticas. Fuente: Elaboración propia.

5.6. Codificación front-end y back-end

En este tema se van a mostrar las dos etapas de la metodología aplicada mas importantes de la aplicación las cuales son el desarrollo y sus pruebas de funcionalidad. Puesto que la aplicación móvil se desarrolla con Java y XML y el sistema monitor con Javascript y PHP se hará mención un código de la programación front-end y back-end de cada uno para que pueda explicarse su función dentro del paquete de código de la aplicación o del sistema monitor.

5.6.1. Etapa 5. Construcción y desarrollo

En esta etapa se va a explicar la codificación de la aplicación y del sistema monitor, se hará mención sobre las funciones que fueron codificadas para que el lector pueda comprender lo que el desarrollador busca lograr en cada parte del código. Muchas de las veces que se realiza la programación de cualquier software el código que se muestra no puede ser comprendido debido a que no se comenta o no está estructurado de forma correcta, por ello el código que se escribe siempre debe estar documentado y comentado para siempre tener presente el por qué de cada línea de código o cada conjunto de líneas.

Primero que nada se va a describir la parte de front-end de la aplicación y del sistema monitor. Como se mencionó en el marco teórico el front-end[13] es la parte de un software que se comunica directamente con un usuario del sistema, podemos llamarlo la interfaz del software. Dentro de la parte front-end de un software o parte frontal se diseña la entrada de datos, es decir todos los formularios, ventanas o pantallas que el usuario va a mirar en su dispositivo y con las que va a interactuar, a través del front-end también es como el software le interpreta sus resultados al usuario por ello estas interfaces deben ser amigables, simples y comprensibles. Después de analizar la parte front-end se va a describir la parte del back-end[13] o parte trasera de un software, esta es la encargada de recibir los datos que el front-end ha recopilado y los va a transformar en información útil para el usuario o para realizar cualquier operación que este programada.

En esta parte del trabajo de investigación únicamente se hará mención de un solo fragmento front-end y back-end de la aplicación y del sistema monitor, esto con motivo de no extender el capítulo y llenarlo de fragmentos interminables de código.

5.6.1.1. Codificación front-end y back-end de la aplicación móvil

Front-end

En la parte de front-end se va a describir el módulo de consultas estadísticas y un fragmento de su código XML el cual de acuerdo a las propiedades establecidas mediante etiquetas hará su transformación para mostrarse como una pantalla interactiva.

A continuación se mostrará un código XML de la actividad de Consultas estadísticas. Activity_estadísticas.xml. Fuente: Elaboración propia.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout
3 xmlns:android="http://schemas.android.com/apk/res/android"
4 xmlns:app="http://schemas.android.com/apk/res-auto"
5 xmlns:tools="http://schemas.android.com/tools"
6 android:layout_width="match_parent"
7 android:layout_height="match_parent"
8 tools:context="tecmartinez.itsmt.Estadística">
9
10 <LinearLayout
11 android:layout_width="match_parent"
12 android:layout_height="wrap_content"
13 android:orientation="horizontal"
14 android:id="@+id/layout_control">
15
16 <fr.ganfra.materialspinner.MaterialSpinner
17 android:id="@+id/ms_seleccion_estadísticas"
18 android:layout_width="250dp"
19 android:layout_height="60dp"></fr.ganfra.materialspinner.MaterialSpinner>
20
21 <Button
22 android:layout_width="120dp"
23 android:layout_height="60dp"
24 android:layout_marginLeft="5dp"
25 android:text="@string/msg_estadísticas_botón_búsqueda"
26 android:background="@color/com_facebook_blue"
27 android:textColor="@color/colorLetrasBotones"
28 android:onClick="BotonGrafica" />
29
30 </LinearLayout>
31
32 <com.github.mikephil.charting.charts.PieChart
33 android:id="@+id/pieChartTec"
34 android:layout_below="@+id/layout_control"
35 android:layout_width="match_parent"
36 android:layout_height="match_parent">
37 </com.github.mikephil.charting.charts.PieChart>
38
39 </RelativeLayout>

```

En el código anterior se puede apreciar los elementos que contiene el formulario como lo son el Spinner con Material Design para las opciones del usuario, el botón que ejecuta la búsqueda de los datos y el lienzo que lleva a cabo la transformación de la agrupación de datos en una gráfica visual coloreada en forma de pay. El resultado se muestra en la Figura 5.8:

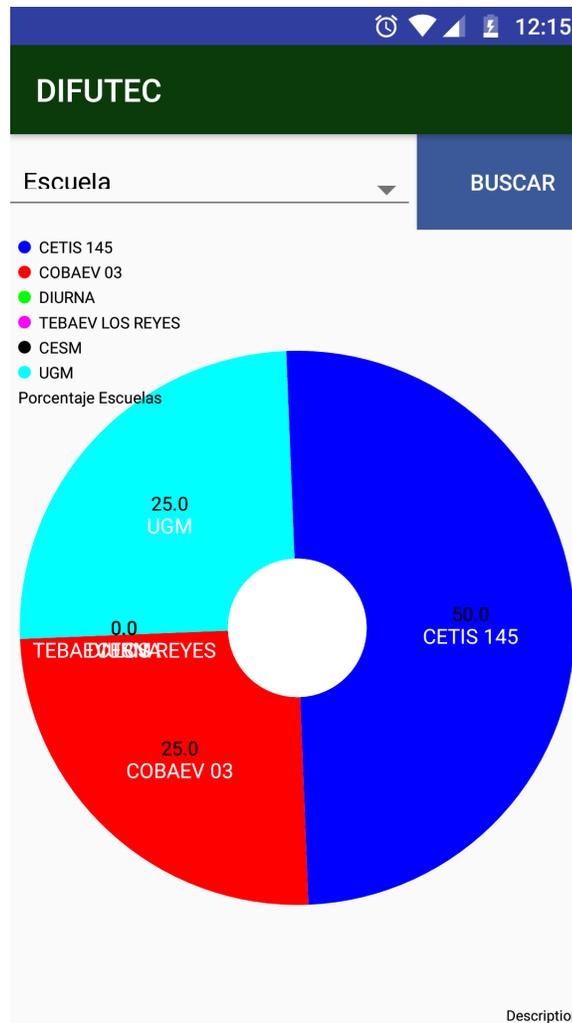


Figura 5.8: Resultado front-end de programación modulo consultas estadísticas en Android. Fuente: Elaboración propia.

Back-end

En la parte de Back-end se va a describir el módulo de consultas estadísticas (del cual ya se hizo mención en la parte de Front-end) y un fragmento de su código JAVA el cual de acuerdo a las propiedades establecidas y palabras reservadas va a construir métodos y funciones que ejecuten la transformación de los datos JSON en algo interpretable para la agrupación de datos de la paquetería de la gráfica.

Primero se llama a las clases que se van a utilizar en variables para que puedan extraer las propiedades de cada una:

Estadística.java. Fuente: Elaboración propia.

```
1 // Lista que va a contener los diferentes tipos de grupos
2 List<String> elementos = new ArrayList<String>();
3
4 //lista de datos recibidos para agrupar
5 List<String> listaDatos = new ArrayList<String>();
6
7 //adaptador y lista que contendra las opciones del spinner
8 List<String> listaOpcionesSpinner = new ArrayList<String>();
9 ArrayAdapter<String> adaptadorOpcionesSpinner;
10
11 //Referencia a la base de datos y a su nodo inicial
12 private FirebaseDatabase midb = FirebaseDatabase.getInstance();
13 private DatabaseReference rootRef = midb.getReference();
14
15 //Gráfica
16 PieChart pieChart;
17
18 ArrayList<Float> valores = new ArrayList<Float>();
19
20 //Spinner con opciones de la gráfica
21 MaterialSpinner msSeleccionGrafica;
```

Después de declarar las variables a utilizar hay que relacionar los componente gráfica y spinner con los que están declarados en nuestro contexto. También a una de las listas declaradas anteriormente hay que añadir las opciones que va a contener el Spinner, una vez añadidas todas las opciones, esa lista se le va a asignar a un adaptador que sera interpretado por el Spinner para que pueda contener valores. Quedando de la siguiente manera:

Estadística.java. Fuente: Elaboración propia.

```
1 protected void onCreate(Bundle savedInstanceState) {
2
3     super.onCreate(savedInstanceState);
4     setContentView(R.layout.activity_estadísticas);
5     pieChart=(PieChart)findViewById(R.id.pieChartTec);
6     msSeleccionGrafica = (MaterialSpinner)findViewById(R.id.
7         ms_seleccion_estadísticas);
8
9     listaOpcionesSpinner.add("Ninguna");
10    listaOpcionesSpinner.add(getText(R.string.msg_lista_spinner_escuela)+"");
11    listaOpcionesSpinner.add(getText(R.string.msg_lista_spinner_carrera)+"");
```

```

11 listaOpcionesSpinner.add(getText(R.string.msg_lista_spinner_localidad)+"");
12 listaOpcionesSpinner.add(getText(R.string.msg_lista_spinner_propuesta)+"");
13 listaOpcionesSpinner.add(getText(R.string.msg_lista_spinner_genero)+"");
14
15 adaptadorOpcionesSpinner = new ArrayAdapter<String>(this, android.R.layout.
    simple_spinner_dropdown_item, listaOpcionesSpinner);
16 adaptadorOpcionesSpinner.setDropDownViewResource(android.R.layout.
    simple_spinner_dropdown_item);
17 msSeleccionGrafica.setAdapter(adaptadorOpcionesSpinner);
18
19 }

```

Para la siguiente parte del código se va a mostrar el apartado de una sola de las opciones ya que todas realizan la misma función de agrupación, comprendiendo una en las demás se realiza lo mismo pero con diferentes nodos de la base de datos, de acuerdo a la agrupación. Para hacer la agrupación primero se hace una lista con todos los diferentes tipos de documentos que contiene la base de datos pero como Firebase Database no contiene ningún método de agrupación de documentos a través de una consulta es necesario realizar la agrupación en el código.

Estadística.java. Fuente: Elaboración propia.

```

1 elementos.clear();
2 listaDatos.clear();
3 rootRef.child(ReferenciasFirebase.PREFERENCIAUSUARIO).addValueEventListener(new
    ValueEventListener() {
4     @Override
5     public void onDataChange(DataSnapshot dataSnapshot) {
6         //Aqui se hace la llamada de todos los nodos de escuela en el nodo
            preferenciausuario
7         for(DataSnapshot carrera:dataSnapshot.getChildren()){
8             elementos.add(carrera.child("escuela").getValue(String.class));
9         }
10
11         //Aqui se realiza la llamada de todos los datos del nodo seleccionado en este
            caso Escuela
12         rootRef.child(ReferenciasFirebase.ESCUELA).addValueEventListener(new
            ValueEventListener() {
13             @Override
14             public void onDataChange(DataSnapshot dataSnapshot) {
15                 for (DataSnapshot carrera:dataSnapshot.getChildren()){
16                     listaDatos.add(carrera.child("siglas").getValue(String.class));
17                 }

```

Una vez que se tengan todos los distintos grupos los cuales están en listaDatos y también los elementos de todos los usuarios los cuales están en la lista elementos hay

que proseguir con la agrupación de todos y cada uno de los diferentes tipos de grupos y a su vez teniendo ya los números de la agrupación obtener el porcentaje de cada uno de los grupos que representa a la cantidad de cada uno. Esto se hace con un ciclo, que cuente cada uno de los diferentes tipos de datos en la lista y almacenando los conteos en un contenedor, para así interactuar con esos totales y realizar una operación de conversión a porcentaje.

Estadística.java. Fuente: Elaboración propia.

```
1 //Se declara un contenedor que va a recibir las cantidades totales de cada grupo
   contado
2 int [] contenedor = new int [listaDatos.size()];
3
4 //se inicializa un ciclo del tamaño de la lista de datos totales o grupos
5 for(int i=0;i<listaDatos.size();i++){
6
7 //al contenedor en la posición actual le asignamos un valor 0 para que tenga
   existencia
8 contenedor[i]=0;
9
10 //Asignamos a una variable llamada contador el elemento actual por agrupar y la
   comparamos con cada uno de los elementos de la
11 //lista de datos total, esto con un for anidado dentro del que ya esta declarado
   anteriormente
12 String comparador = listaDatos.get(i);
13 for (int j=0;j<elementos.size();j++){
14 if(elementos.get(j).toString().equals(comparador)){
15 contenedor[i]++;
16 }
17 }
18 }
19
20 //A continuación hay que declarar un total acumulativo que nos pueda ayudar a
   realizar la suma para la operación de conversión a porcentaje
21 int total = 0;
22 for(int i=0;i<contenedor.length;i++){
23 total = total + contenedor[i];
24 }
25
26 //una vez se tenga el total, se va a declarar un arreglo de flotantes donde se
   almacenaran los porcentajes de cada una de las opciones de la agrupación
27 float [] contenedorPorcentaje = new float [contenedor.length];
28 for(int i=0;i<contenedor.length;i++){
29 contenedorPorcentaje[i] = (contenedor[i] * 100) /total;
30 valores.add(contenedorPorcentaje[i]);
31 }
```

En el siguiente fragmento de código se muestra como los datos obtenidos van a interactuar con la gráfica y sus métodos para así conformarla en el lienzo. Lo que se debe realizar a continuación es que con la variable declarada al inicio de la clase pieChart se le implementaran algunos métodos que harán mas interactiva la gráfica de pay para el usuario. Ademas se integraran la serie de datos que se obtuvieron en las listas anteriores, solo los datos de porcentaje y también las etiquetas de cada uno en el mismo orden que fueron calculados ya que no van relacionados de alguna manera.

Estadística.java. Fuente: Elaboración propia.

```

1  pieChart.setRotationEnabled(true); //se habilita la rotacion de la gráfica
2  pieChart.setHoleRadius(25f); //se enviá el radio del circulo de la gráfica
3  pieChart.setTransparentCircleAlpha(0); //se le da un estado solido al color
4  pieChart.setCenterTextSize(10); //tamaño de letra 10 en el centro
5
6  ArrayList<PieEntry> yEntrys = new ArrayList<>(); //se declara una lista que va a
   contener los porcentajes
7  ArrayList<String> xEntrys = new ArrayList<>(); //se declara una lista con los
   nombres de cada serie de datos
8
9  //a la lista de datos de porcentajes se llena con los datos obtenidos en el
   fragmento de código anterior a través un ciclo
10 for(int i = 0; i < contenedorPorcentaje.length; i++){
11  yEntrys.add(new PieEntry(contenedorPorcentaje[i], listaDatos.get(i)));
12 }
13
14 //lo mismo para el caso de los nombres de las series, solo que a diferencia de
   obtener los valores numericos se van a extraer todos los nombres de la
   consulta de agrupación
15 for(int i = 0; i < listaDatos.size(); i++){
16  xEntrys.add(listaDatos.get(i));
17 }
18
19 //Agregar los datos de porcentaje a los vectores creados anteriormente
20 PieDataSet pieDataSet = new PieDataSet(yEntrys, "Porcentaje Escuelas");
21 pieDataSet.setSliceSpace(2);
22 pieDataSet.setValueTextSize(12);
23
24 //Agregar colores diferentes a la gráfica
25 ArrayList<Integer> colors = new ArrayList<>();
26 colors.add(Color.BLUE);
27 colors.add(Color.RED);
28 colors.add(Color.GREEN);
29 colors.add(Color.MAGENTA);
30 colors.add(Color.BLACK);
31 colors.add(Color.CYAN);
32 colors.add(Color.DKGRAY);

```

```
33 colors.add(Color.YELLOW);
34 pieDataSet.setColors(colors);
35
36 //Poner una leyenda de texto en la gráfica
37 Legend legend = pieChart.getLegend();
38 legend.setForm(Legend.LegendForm.CIRCLE);
39 legend.setPosition(Legend.LegendPosition.LEFT_OF_CHART);
40
41 //Crear el objeto e inicializarlo en el lienzo
42 PieData pieData = new PieData(pieDataSet);
43 pieChart.setData(pieData);
44 pieChart.invalidate();
```

Anteriormente se puso todos los códigos dentro de un if, para cada una de las diferentes opciones cargadas en el Spinner. De esta manera solo se cambian los nodos de búsqueda en la base de datos y se obtiene la información agrupada a través de un algoritmo de ordenamiento y poder obtener los cálculos necesarios para elaborar la gráfica.

5.6.1.2. Codificación front-end y back-end del sistema monitor web

Front-end

En el front-end del sistema monitor se incluirá solo un pequeño fragmento de uno de los documentos HTML del proyecto que se generaron para dar la comunicación directa con el usuario, el sitio web esta diseñado con frameworks como Bootstrap los cuales dan un plus de diseño y también a la experiencia de usuario gracias a la implementación de animaciones y muchas mas funciones como ventanas modales y elementos con estilos muy atractivos. La comunicación con el usuario o recopilador de información front-end debe darse en una interfaz simple y minimista para la comprensión del usuario y poder trabajar con las opciones para la base de datos que se reflejaran en la aplicación móvil. El código HTML que se escribe a continuación da como resultado a la pantalla inicial del usuario monitor o index, desde donde también puede visualizar las opciones de la gráfica como acción principal en la pagina o seleccionar alguna de las opciones en el menú de arriba.

index.php. Fuente: Elaboración propia.

```
1 </head>
2 <body>
3 <script src="js/responsive.js"></script>
4 <script src="js/login.js"></script>
```

```

5 <nav class="navbar navbar-inverse navbar-fixed-top">
6 <div class="container">
7 <div class="navbar-header">
8 <a class="navbar-brand navbar-link" href="#">
  <strong>DIFUTECH</strong></a>
9 <button class="navbar-toggle collapsed" data-toggle="collapse" data-target="#
  navcol-1"><span class="sr-only">Toggle navigation</span><span class="icon-
  bar"></span><span class="icon-bar"></span><span class="icon-bar"></span></
  button>
10 </div>
11 <div class="collapse navbar-collapse" id="navcol-1">
12 <ul class="nav navbar-nav navbar-right">
13 <li role="presentation"><a onclick="irMonitor()">MONITOR </a></li>
14 <li role="presentation"><a onclick="irTest()">TEST EDUCATIVO </a></li>
15 <li role="presentation"><a onclick="irCarrera()">CARRERAS </a></li>
16 <li role="presentation"><a onclick="irEscuela()">ESCUELAS</a></li>
17 <li role="presentation"><a onclick="irUsuarios()">USUARIOS</a></li>
18 <li role="presentation"><a onclick="irProyectos()">PROYECTOS</a></li>
19 <li role="presentation"><a onclick="irLocalidades()">LOCALIDADES</a></li>
20 <li role="presentation" class="dropdown">
21 <a href="#" class="dropdown-toggle" data-toggle="dropdown">OPCIONES<b class="
  caret"></b></a>
22 <ul class="dropdown-menu">
23 <li><a role="menuitem" tabindex="-1" data-toggle="modal" data-target="#
  modalReautenticarUsuario">CAMBIAR CLAVE</a></li>
24 <li><a role="menuitem" tabindex="-1" data-toggle="modal" data-target="#
  modalReautenticarUsuarioEliminar">BORRAR CUENTA</a></li>
25 <li><a role="menuitem" tabindex="-1" onclick="salir()">CERRAR SESION</a></li>
26 </ul>
27 </li>
28 </ul>
29 </div>
30 </div>
31 </nav>
32 <div class="panel panel-info" id="margen">
33 <div class="panel-heading">OPCIONES DE GRAFICA</div>
34 <div class="panel-body">
35 <center>
36 <div class="form-group" id="margenform">
37 <label class="" for="generoUser">Opciones</label>
38 <select style="max-width: 300px" id="opcionesGrafica" class="opcionesGrafica
  form-control" name="generoUser">
39 <option value="">Selecciona una opción</option>
40 <option value="Escuela">Escuela</option>
41 <option value="Carrera">Carrera</option>
42 <option value="Localidad">Localidad</option>
43 <option value="Propuesta">Propuesta</option>
44 <option value="Genero">Genero</option>
45 </select>

```

5.6 Codificación front-end y back-end

```
46 </div>
47 <button type="button" class="btn btn-info" onclick="mostrarGrafica()">MOSTRAR</
    button>
48 </center>
49 </div>
50 </div>
51 <div id="graficaCircular" style="width: 100%; height: 500px; margin: 0 auto"></
    div>
52 <?php include("modal/modalReautenticarUsuario.php"); ?>
53 <?php include("modal/modalReautenticarUsuarioEliminar.php"); ?>
54 <script src="js/jquery.min.js"></script>
55 <script src="bootstrap/js/bootstrap.min.js"></script>
56 <script src="js/accionesmenu.js"></script>
57 <script src="js/appIS.js"></script>
58 <script src="js/currentUserOptions.js"></script>
59 </body>
60 </html>
```

El resultado del código que esta anteriormente se muestra en la Figura 5.9.

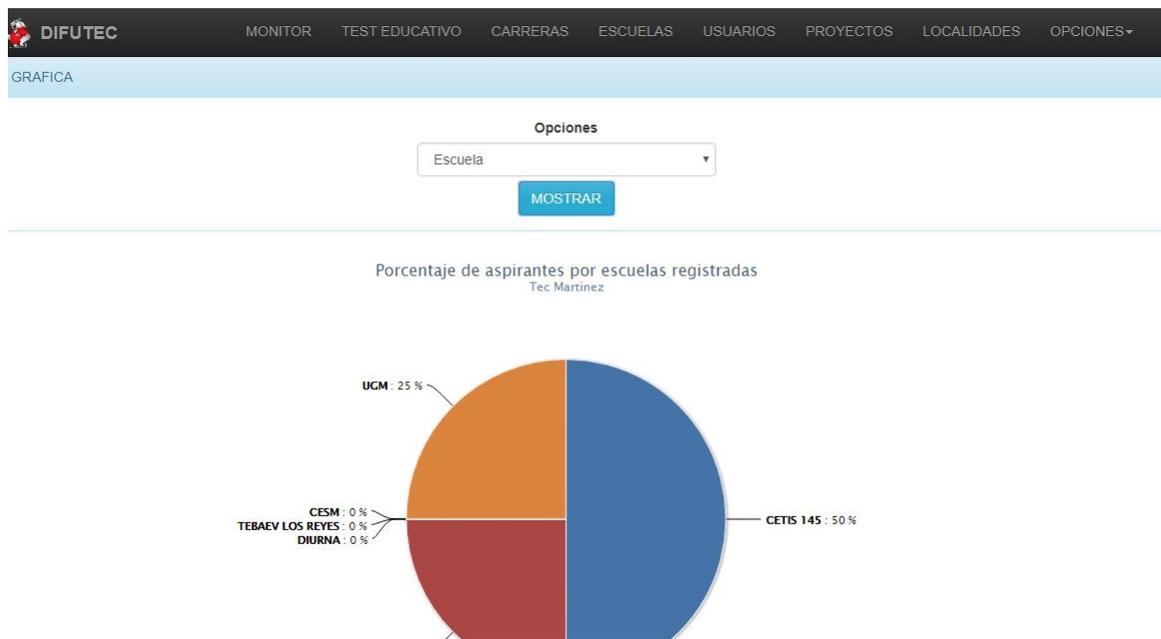


Figura 5.9: Resultado del archivo index.php el cual contiene la maquetacion o codificación front-end del sistema monitor. Fuente: Elaboración propia.

En la página mostrada en la Figura 5.9 se contiene con un select que maneja las opciones para visualizar la gráfica, estas opciones son las mismas que la aplicación

móvil, esto con el motivo de que los usuarios del sistema monitor también puedan obtener resultados de las opciones que han cargado en la base de datos.

Back-end

Como se mencionó en el capítulo del marco teórico, la parte back-end de un software es la parte trasera o la que se encarga de recibir los datos que un usuario envía y los transforma en información útil para el mismo y para el sistema, aunque en el ámbito de la programación web, la parte del back-end se refiere a todas las funciones que son ejecutadas en el lado del servidor. Pero en este caso, la mayor parte del sistema monitor web y sus funciones fueron programadas de lado del cliente con Javascript, debido a que las tecnologías de Firebase solo permiten comunicación a través de este lenguaje.

Aunque la comunicación se genera con Javascript no quiere decir que no haya funciones programadas del lado del servidor, aunque son escasas, estas son importantes ya que algunas librerías como lo son HighCharts que es la librería con la que se genera la Gráfica requieren tanto de Javascript como de PHP.

Por todo lo mencionado anteriormente se va a establecer que cualquier función programada para procesar datos y enviarlos a Firebase Database aun siendo ejecutada del lado del cliente se va a considerar como back-end ya que será una función encargada de trabajar los datos que se reciban en la interfaz.

El archivo que se va a describir a continuación contiene una de las funciones para generar la gráfica en caso de ser elegida la opción de Escuela en el select. Al igual que la aplicación móvil ejecuta una consulta a la base de datos de Firebase a través de una referencia en el nodo raíz, después de ubicar la referencia en el nodo raíz hay que profundizar en los niveles deseados para encontrar los nodos que cumplan con la búsqueda y de esta manera obtener los datos deseados y agruparlos para que se puedan hacer las operaciones necesarias para construir la gráfica.

Script dentro del archivo `index.php`. Fuente: Elaboración propia.

```
1 //se crea una referencia basica de la base de datos y se ubica en el nodo raíz
   con el método ref
2 var rootRef = firebase.database().ref();
3
4 //se profundiza la búsqueda en los niveles siguientes para encontrar al nodo que
   se busca en este caso es preferenciasUsuario y busca los datos de escuela
   en cada usuario
```

5.6 Codificación front-end y back-end

```
5 referenciaEscuela = rootRef.child('preferenciasUsuario');
6
7 //Se hae una referencia que parta de la referencia anterior hacia el nodo que
  contiene el dato de escuelas registradas
8 referenciaEscuelasTotal = rootRef.child('escuela');
9
10 //se crean los dos arrays que van a contener los datos y la cantidad de
    diferentes grupos existentes para hacer el mismo algoritmo de agrupación que
    en la aplicación
11 var listaDatos = new Array();
12 var elementos = new Array();
13
14 //a través de un ciclo for se obtienen los datos a través de un Snapshot o
    captura de información, una vez obtenida con el ciclo se recorren los datos
    de acuerdo al nodo deseado.
15 referenciaEscuela.once('value', function(snapshot) {
16 snapshot.forEach(function(childSnapshot) {
17
18 var childKey = childSnapshot.key; //nodo de llave UID
19 var childData = childSnapshot.val(); //nodo de la información solicitada en este
    caso escuela
20 elementos.push(childData.escuela); //se agrega a la lista el nodo obtenido en su
    parametro nombrado igual que en la base de datos.
21 });
```

Con el código anterior se obtienen todos los datos necesarios para crear la agrupación, es decir todos los datos de los usuarios y la opción que han elegido en cuanto la preferencia de escuela de procedencia. Entonces ahora lo que prosigue es obtener los datos para comparar es decir, la cantidad total diferente de escuelas registradas en la base de datos que al igual que con la consulta anterior sera en base de una referencia a un nodo, también se extraerán los datos con un Snapshot y un ciclo.

Script dentro del archivo index.php. Fuente: Elaboración propia.

```
1 //Comienza la referencia en base a la referencia root o raíz
2 referenciaEscuelasTotal.once('value', function(snapshotNew){
3
4 //se declara el snapshot que va a contener los datos de la consulta
5 snapshotNew.forEach(function(childSnapshotNew){
6
7 //se declara una variable con contiene la información del snapshot
8 var datos = childSnapshotNew.val();
9
10 //se obtienen los datos del nodo siglas y se añaden a la lista
11 listaDatos.push(datos.siglas);
12 });
13
```

```

14 //se declara un contenedor que va almacenar los conteos de cada grupo
15 var contenedor = [];
16
17 //iniciamos un for comparando cada uno de los datos de una lista con los de la
    otra
18 for(var i=0;i<listaDatos.length;i++){
19 contenedor[i]=0;
20 var comparador = listaDatos[i];
21 for(var j=0;j<elementos.length;j++){
22 if(elementos[j]==comparador){
23 contenedor[i]++;
24 }
25 }
26 }

```

Al finalizar la agrupación de los datos es necesario iniciar la conversión a porcentaje de cada uno de los conteos de los grupos almacenados en el contenedor que se declaro anteriormente y se envían a la gráfica.

Script dentro del archivo index.php. Fuente: Elaboración propia.

```

1 var contenedorPorcentaje = []; //vector que va almacenar los porcentajes
2 var total = 0; //total del conteo de todos los datos
3 for(var i=0;i<contenedor.length;i++){
4 total = total + contenedor[i]; //se van sumando todos los grupos para obtener un
    total
5 }
6
7 // se realiza la regla de tres para cada grupo
8 for(var i=0;i<contenedor.length;i++){
9 contenedorPorcentaje[i] = (contenedor[i] * 100) /total;
10 }
11
12 var datos=""; //se declara una variable que va a contener los valores de la
    comparación
13
14 //se realiza un for para concatenar todos los datos como texto y enviarselos a
    la librería de la
15 for(var i=0;i<contenedorPorcentaje.length;i++){
16 //se van concatenando los resultados y también los nombres de cada grupo
17 datos+="["+listaDatos[i]+"', "+contenedorPorcentaje[i]+"], ";
18 }
19 //a través de la tecnologia de ajax se llama a un documento llamado
    graficaEscuela.php y se le enviá la variable concatenada en una asignacion
    JSON que es variableContenedor
20 $("#graficaCircular").load("control/graficaEscuela.php",{variableContenedor:
    datos});

```

Anteriormente después de realizar la conversión de los datos, estos fueron concatenados en una variable de texto para que fueran enviados a un php que va a construir la gráfica de manera dinámica.

graficaEscuela.php. Fuente: Elaboración propia.

```
1 <?php
2 //se declara una variable que recibe los datos atraves del método REQUEST y
   buscamos la variable nombrada como variableContenedor
3 $variable = $_REQUEST['variableContenedor'];
4
5 //se imprime una línea de código que va a construir la gráfica y todas las
   configuraciones para mostrar los datos
6 echo "<script>chart = new Highcharts.Chart({
7 chart: {
8   renderTo: 'graficaCircular'
9 },
10 title: {
11   text: 'Porcentaje de aspirantes por escuelas registradas'
12 },
13 subtitle: {
14   text: 'Tec Martinez'
15 },
16 plotArea: {
17   shadow: null,
18   borderWidth: null,
19   backgroundColor: null
20 },
21 tooltip: {
22   formatter: function() {
23     return '<b>'+ this.point.name + '</b>: '+ this.y + ' %';
24   }
25 },
26 plotOptions: {
27   pie: {
28     allowPointSelect: true,
29     cursor: 'pointer',
30     dataLabels: {
31       enabled: true,
32
33       color: '#000000',
34       connectorColor: '#000000',
35       formatter: function() {
36         return '<b>'+ this.point.name + '</b>: '+ this.y + ' %';
37       }
38     }
39   }
40 },
41 series: [{
```

```

42 type: 'pie',
43 name: 'Browser share',
44 data: [ ".$variable." ]
45 }
46 });</script>;
47 //Aquí finaliza la construcción de los datos que fueron enviados en forma de
    cadena
48 ?>

```

El resultado como se puede apreciar es el de la figura 4.8 en petición del usuario en la opción de escuela.

5.6.2. Etapa 6. Pruebas y evaluación

Una vez construida la aplicación, es necesario probar su funcionamiento en diferentes dispositivos en los cuales la aplicación pueda mantener su diseño original, sin presentar deformaciones en sus elementos o que la funcionalidad se vea afectada en dispositivos diferentes. Para realizar pruebas de la aplicación se hizo una tabla mostrada en el Cuadro 5.4 que califica el desempeño de la aplicación en dispositivos con diferentes dimensiones y diferentes versiones del sistema operativo superiores a las establecidas para las características del dispositivo.

Modelo Dispositivo	¿Presenta deformación en el diseño?	¿Realiza todas las funciones de manera aceptable?	¿Presentó fallas o lentitud al ejecutar la aplicación?	Calificación final	Versión S.O.
Moto Z2 Play	No	Si	No	Muy Aceptable	7.1
Huawei G Elite	No	Si	No	Muy Aceptable	5.0
LG X Style	No	Si	No	Muy Aceptable	6.0
LG Leon	No	Si	Si	Poco Aceptable	4.1
Sony Xperia E5	No	No	No	Aceptable	5.1

Cuadro 5.4: Tabla de calificación y pruebas de la aplicación en diferentes dispositivos.

Fuente: Elaboración propia.

La conclusión por parte del desarrollador es que es una aplicación que esta correctamente programada y diseñada para su objetivo, sol hace falta realizar una prueba piloto para ver como se desempeña en un ámbito real y con muchos usuarios haciendo peticiones desde la aplicación al mismo tiempo.

5.7. Implementación de la app desarrollada

Una vez que al codificación de la aplicación fue terminada lo que prosigue es generar el archivo instalable para que los dispositivos con sistema operativo android puedan instalar la app sin problemas. Al hablar de una aplicación no podemos hablar directamente de una implementación de un sistema de software a un proceso sino de un pequeño programa que se ejecute en un dispositivo en cualquier lugar. Considerando la premisa anterior es necesario considerar la cantidad de usuarios que utilizaran la aplicación y la cantidad de peticiones de información. Por ello en este tema se va a explicar la culminación real del proyecto, sus características en cuanto a la aplicación y al sistema monitor web.

5.7.1. Etapa 7. Liberación

En este apartado corresponde a la generación de la aplicación en base a sus codificaciones front-end y back-end. El resultado es una aplicación que, hablando en términos de programación esta finalizada, el archivo apk de instalación y el proyecto han sido entregados a la institución para que se decida sobre la mejor opción de distribución.

5.7.1.1. Requisitos de la app

Esta aplicación es instalable en cualquier dispositivo que cumpla los siguientes requisitos:

1. Sistema operativo Android Version 5.0 o superior.
2. Espacio disponible para archivo apk de instalación 12.15MB.
3. Espacio disponible para aplicación y datos 27.38MB.
4. Acceso a Internet

5. Memoria RAM 1 GB.
6. Memoria caché 2MB.

5.7.1.2. Requisitos del sistema monitor web

1. Acceso a Internet
2. Tener acceso a un explorador web compatible: Google Chrome, Firefox, Edge, Safari.

5.7.1.3. Manuales de usuario

Para la sección de manuales de usuario el trabajo esta pensado para dividirse de la siguiente manera:

1. El sistema monitor web contara con un manual de trabajo y sus funciones ademas de una capacitación para su uso y trabajo.
2. La aplicación web contara con un apartado de ayuda e instrucciones básicas de uso en sus funciones en la pagina del ITSMT.

Conclusión

Todas las tecnologías que fueron utilizadas para el desarrollo de la aplicación, los algoritmos utilizados y la manera de estructurar el código se debe a los conocimientos que se adquieren en base a la documentación de las mismas tecnologías. Muchas de estas son innovadoras y como tal, no se tiene alguna documentación de su uso para procesos exactos, si no recopilaciones de desarrollos y usos de los métodos de cada una de manera general, por ello muchos de los procesos cuentan con estructuras básicas las cuales fueron extraídas de la documentación oficial de cada proveedor de las tecnologías elegidas, ejemplo Google Firebase, Twitter, Facebook, Highcharts, etc. Los algoritmos que se desarrollaron estan basados en la documentación aunque sufrieron modificaciones, un ejemplo de esto es que Firebase Database es una base de datos que recientemente se desarrollo y no cuenta con muchas funcionalidades o metodos de consulta, por ello para la agrupación de datos se utilizó un algoritmo desarrollado por el programador de

5.7 Implementación de la app desarrollada

la aplicación, el cual tiene como objetivo hacer un método de agrupación sin recurrir a la base de datos.

Procesos como los que se mencionaron anteriormente se encuentran mucho en este capítulo, pues basados en los primeros diagramas y las peticiones del ITSMT para el objetivo de la aplicación fue necesario que se hiciera un desarrollo de forma muy definida, es decir, los algoritmos que se utilizan solo cumplen su cometido y sería difícil implementarlos en otros procesos.

6 Experimentación y resultados

A continuación se presenta una explicación de las pruebas que se realizaron para calificar el funcionamiento de la aplicación y su desempeño a mayor escala. Las pruebas que aquí se muestran fueron observadas y analizadas cuidadosamente para verificar que la aplicación cumpliera los objetivos y las expectativas del cliente. A continuación se mostraran las pruebas y herramientas utilizadas para calificar la experiencia del usuario con la aplicación. En este proyecto se utilizaron varias técnicas de recopilación de información para probar el funcionamiento de la aplicación pero para calificar la experiencia del usuario y verificar si se cumplen las hipótesis planteadas se utilizo solo una y en un grupo de personas que hicieron uso de la aplicación.

6.1. Actividad de recolección y análisis de datos

La actividad de recolección de datos consiste en una encuesta de 10 enunciados, esta encuesta fue hecha a los 30 participantes de 6to semestre de una institución educativa preparatoria. Esta encuesta fue realizada después de explicarles el uso y objetivo de la aplicación móvil DIFUTEC. Esta encuesta consiste en enunciados con 3 opciones de calificación las cuales son nada satisfecho, satisfecho, muy satisfecho. Después de recopilar las 30 encuestas, se hizo el conteo de los datos y se realizó una gráfica que represente los datos obtenidos de cada pregunta.

6.1.1. Enunciado 1

Los colores de la aplicación son agradables y dan buena apariencia.

El primer enunciado tiene como objetivo identificar el impacto de la aplicación en una primera impresión de los usuarios que la utilizaron. El diseño y todos los colores que la aplicación tiene son muy importantes pues se utilizaron algunos de los colores Institucionales del ITSMT los cuales son verde y blanco, se añadió en algunos botones un color azul que pueda resaltar siempre la opción principal o en cada pantalla y este



Figura 6.1: Gráfica interpretativa enunciado 1. Fuente: Elaboración propia.

color es muy contrastado a los de la aplicación por lo que se considera llamativo para el usuario. Como se aprecia en la gráfica de la Figura 6.1, solo el 7% de la muestra no está de acuerdo o nada satisfecho con los colores elegidos, algunos de los comentarios de los usuarios es que debe incluir más colores la aplicación. El 83% y el 10% son la escala de satisfecho y muy satisfecho respectivamente, estos usuarios encuestados encuentran atractiva la combinación simple de 3 colores en la aplicación dando un toque institucional sencillo y funcional a la misma.

6.1.2. Enunciado 2

El menú principal es comprensible y es fácil navegar entre sus opciones

En el segundo enunciado el objetivo es identificar si la pantalla principal de la aplicación tiene un diseño que los usuarios pueden comprender de manera fácil o es complicada. Esto es un punto importante ya que la pantalla principal de una aplicación y su menú de opciones serán la pantalla que de la bienvenida al usuario y le proporcione rutas a cada una de las funciones de la aplicación. En la gráfica de la Figura 6.2 se aprecia como ningún usuario que probó la aplicación seleccionó la opción de nada satisfecho, mientras que satisfecho y muy satisfecho tuvieron resultados del 27% y 73% respectivamente. Por lo que se considera que el menú tiene muy buena aceptación y es fácil de entender.



Figura 6.2: Gráfica interpretativa enunciado 2. Fuente: Elaboración propia.

6.1.3. Enunciado 3

La función de Nota Tec es muy útil para conocer noticias sobre el ITSMT a través de la red social Twitter

El objetivo del enunciado 3 es verificar que una de las herramientas que se utilizaron como línea de tiempo de noticias, en este caso Twitter y su timeline sean ampliamente aceptados por los usuarios de la aplicación puesto que es una red social con información confiable y mas segura que Facebook. Por ello se utilizo su timeline o línea de tiempo vinculada a una cuenta que se hizo para el ITSMT con la cual podrán publicar información y noticias importantes sobre muchas de las actividades de la institución. En la gráfica de la Figura 6.3 se puede observar que ningún usuario que probo la aplicación se encuentra en el rango de no satisfecho y muchas de las observaciones señalan que es bueno poder visualizar una línea de tiempo de Twitter sin tener una cuenta registrada en la red social, la serie de satisfecho y muy satisfecho tuvieron un porcentaje de 40% y 60% respectivamente. Esto indica que a la mayoría de los usuarios les agrada la idea de conocer y estar informados de las actividades de la institución a través de esta red social.

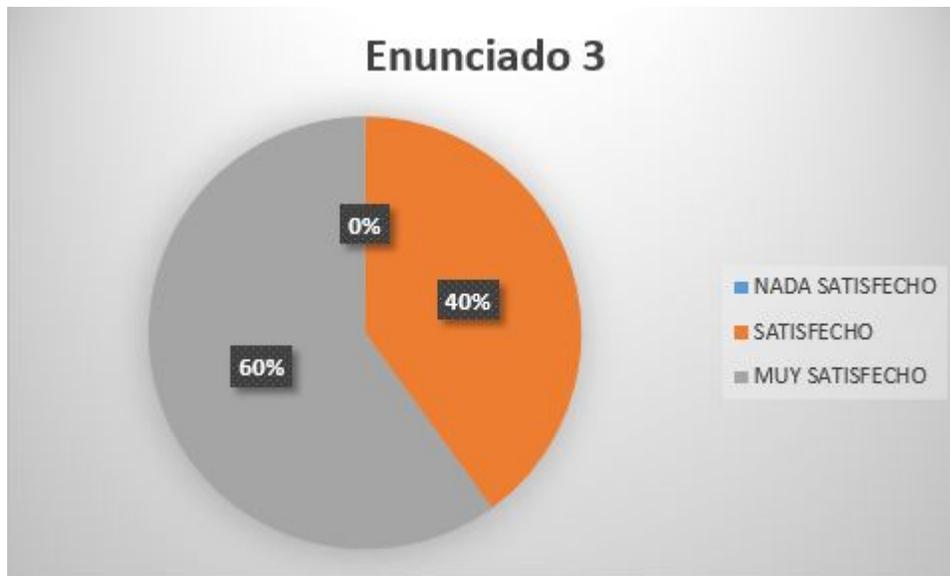


Figura 6.3: Gráfica interpretativa enunciado 3. Fuente: Elaboración propia.

6.1.4. Enunciado 4

Los formularios de inicio de sesión y registro de preferencias son entendibles e intuitivos.

En este enunciado el objetivo principal es identificar la fluidez con que un usuario que recién instala la aplicación puede relacionarse o integrarse de manera que sea agradable a él. Cuando la aplicación se ejecuta sin tener ningún usuario, ofrece la opción de registrarse mediante dos diferentes opciones las cuales son mediante un correo electrónico o un id de cuenta de facebook. El inicio de sesión de un usuario nuevo lleva directo a un formulario de registro de preferencias para que el usuario solo una vez tenga que registrar las preferencias educativas que tiene. En la Figura 6.4 el 13% de usuarios de la muestra no están satisfechos con este proceso, algunas de las observaciones que algunos de estos participantes señalaron fueron que no les parece funcional que haya un formulario de petición de datos justo en el registro de sus credenciales. Un 44% de los participantes señalan que es satisfactorio o aceptable y el 43% de los participantes encuestados están muy satisfechos, señalando que es muy útil que solo una vez tengan que registrar preferencias en la aplicación ya que de hacerlo en el futuro perderían tiempo registrándolas u omitiéndolas.

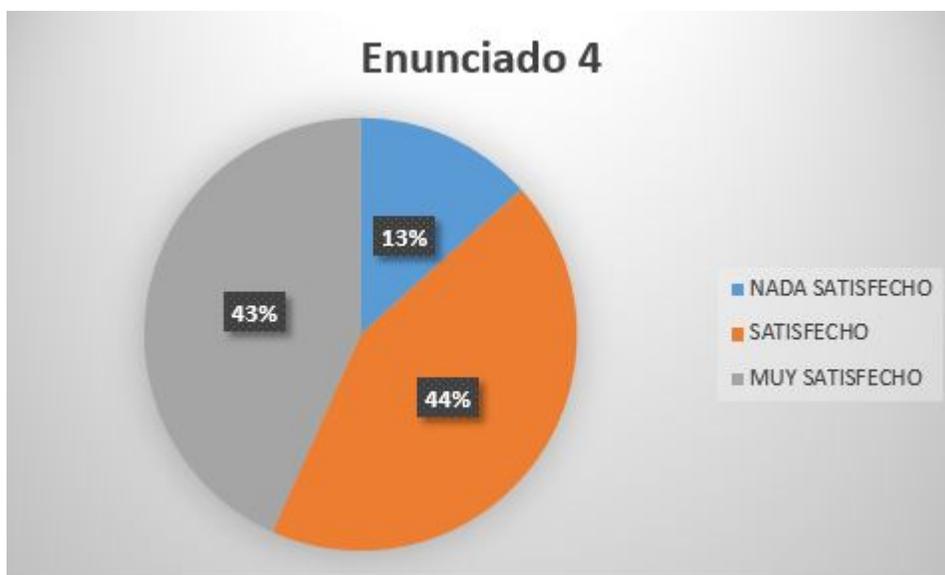


Figura 6.4: Gráfica interpretativa enunciado 4. Fuente: Elaboración propia.

6.1.5. Enunciado 5

Los formularios de orientación educativa son intuitivos y bien estructurados

El test de orientación educativa y su nivel de aceptación se analiza en este enunciado. Las preguntas o enunciados que contiene son elaborados por el desarrollador de la aplicación mediante un estudio de las aptitudes y actitudes que conforman a los alumnos de distintas carreras educativas. Como se muestra en la Figura 6.5 los alumnos encuestados encuentran que un test de orientación educativa integrado en una aplicación es muy atractivo, aunque un 7% de la muestra encuestada señala que los resultados que ofrece el test deberían ser globales en lugar de estar centrados en las carreras que oferta el ITSMT, el 50% está satisfecho con el proceso aunque una observación resalta más en este grupo la cual fue “Un test está bien para ayudarme a elegir que debo estudiar pero sería interesante que también me dijera donde se puede realizar la carrera que de como resultado”. Observaciones como esta son críticas constructivas para mejoras de la aplicación en un futuro. Por otro lado el 43% de los encuestados está muy satisfecho pues varios de ellos señalan que “la aplicación debe atraer al estudiante a la institución”.



Figura 6.5: Gráfica interpretativa enunciado 5. Fuente: Elaboración propia.

6.1.6. Enunciado 6

La autenticación de usuario con la red social Facebook me parece muy útil

Para identificar si el inicio de sesión con una cuenta de Facebook es realmente útil se utilizó este enunciado, el cual señala si lo anterior se cumple o no. La opción de registro o inicio de sesión con una cuenta de Facebook se realiza para aquellos usuarios que desean ahorrar tiempo y esfuerzo en llenar el formulario ya que, la aplicación extrae los datos públicos de la cuenta y los envía a la base de datos para tener un acceso seguro de Facebook mediante un token en la aplicación. De acuerdo a la gráfica de la Figura 6.6 solo el 3% de la aplicación no está satisfecho puesto que prefieren guardar o abstenerse de brindar sus cuentas en una aplicación diferente de Facebook. El otro 44% y 53% son satisfecho y muy satisfecho respectivamente, estos usuarios encuestados aceptan y les parece más útil un inicio o registro a través de una red social como Facebook pues ahorran cierto tiempo en llenar formularios.

6.1.7. Enunciado 7

Las gráficas en el apartado de datos estadísticos son entedibles y diseñadas correcta-



Figura 6.6: Gráfica interpretativa enunciado 6. Fuente: Elaboración propia.

mente.

El objetivo de este enunciado es saber si las gráficas que se generan en la aplicación son entendibles por los usuarios. De acuerdo a la Figura 6.7 solo el 7% de la muestra encuestada no está satisfecha con las gráficas que se muestran pues consideran que no es de utilidad conocer las preferencias de otros usuarios, por otro lado el 60% de los encuestados que eligieron satisfecho, señalando que la gráfica es de mucha utilidad y debe tener un seguimiento pues al igual que el 33% de los usuarios que eligieron muy satisfecho, están de acuerdo que es más fácil interpretar los datos de otros usuarios en forma de gráfica sin conocerlos realmente.

6.1.8. Enunciado 8

El tiempo de respuesta de la aplicación en sus funciones me parece perfecto.

Debido a la gran incorporación de librerías externas a Android, fue necesario evaluar el tiempo de respuesta de la aplicación en todas sus funciones. El objetivo de este enunciado es verificar que los tiempos de respuesta que tiene la aplicación sean aceptados por el usuario o si deben sacrificarse algunas funciones en beneficio del rendimiento. Como se muestra en la Figura 6.8, por primera vez un número considerable de los usuarios encuestados el cual es el 23% de estos, señalan que el tiempo de respuesta no



Figura 6.7: Gráfica interpretativa enunciado 7. Fuente: Elaboración propia.

es nada satisfactorio realizando la observación de que “esperar mas de 5 o 7 segundos en una aplicación ya es muy chocante”, esto no quiere decir que la aplicación siempre funcione lento, simplemente que hay dispositivos que parece ser no adaptan bien las tecnologías que se implementaron en la aplicación como las TimeLine de Twitter o la autenticación con Facebook. El 57% de los encuestados señalan que el tiempo es adecuado para una primer versión y solo el 20% afirma que son los tiempos correctos en respuesta a una solicitud con tal cantidad de datos.

6.1.9. Enunciado 9

Los diseños en los apartados de oferta académica y proyectos tec son intuitivos y es fácil de interpretar la información recibida.

Una de las funciones principales de la aplicación es dar publicidad a la oferta académica y para verificar que la manera en la que esta funcionando este proceso en la aplicación sea la correcta, se hizo un enunciado que calificara el nivel de aceptación y satisfacción por parte del usuario. Este es uno de los enunciados que mas importan puesto que la aplicación tiene un rol de herramienta de promoción o publicidad y la información que enviá debe ser estructurada y resumida de tal forma que de un impacto y sea atractiva para el usuario. De acuerdo a la Figura 6.9 el 10% de los usuarios no están satisfechos

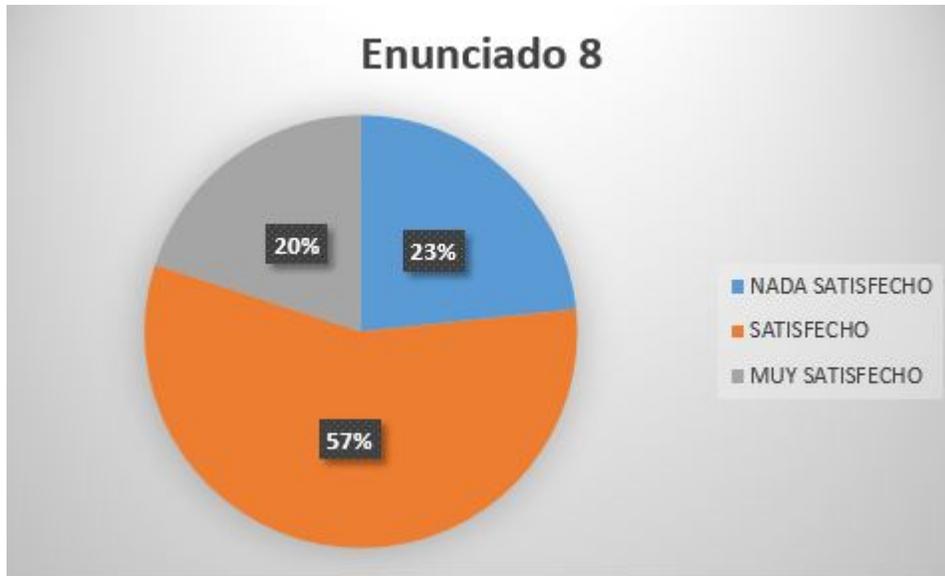


Figura 6.8: Gráfica interpretativa enunciado 8. Fuente: Elaboración propia.

con el diseño, algunos de estos usuarios mencionan que “enlistar las carreras y darles un resumen no siempre da un conocimiento de lo que es en realidad” un ejemplo de lo que estos usuarios quieren decir es que confunden el elegir Sistemas Computacionales con “arreglar computadoras” o Gestión Empresarial con “pura contabilidad”. El 50 % de los encuestados señalan que el diseño y la información recibida es muy atractivo pero hay observaciones de “mejoras en el diseño” para que se vea todavía mas atractivo y amigable. El 40 % de los encuestados han señalado que la estructura e información combinados con el diseño es la mejor opción y se sienten totalmente satisfechos con esta funcionalidad o modulo de la aplicación.

6.1.10. Enunciado 10

Los estilos del diseño y la funcionalidad hacen que la aplicación sea intuitiva, útil y agradable de utilizar.

Despues de evaluar individualmente las funciones de la aplicación con ayuda de los usuarios que participan en la actividad, es necesario conocer el nivel de aceptación de la aplicación de manera general y ese es el objetivo del enunciado 10, calificar la aplicación de forma que combinando el diseño y la funcionalidad, sea total o parcialmente agradable al usuario de acuerdo a la usabilidad de esta. De acuerdo a la Figura 6.10, el



Figura 6.9: Gráfica interpretativa enunciado 9. Fuente: Elaboración propia.

10 % de la aplicación no está satisfecho con el desempeño artístico de la misma pues señalan que el diseño es muy simple y podría mejorarse aun más. El 47 % señala que están satisfechos con el desempeño de las funciones y módulos de la aplicación, sin embargo, señalan que el diseño aunque ya es agradable y le da a la aplicación una gran fluidez puede ser aun más atractivo. El 43 % de la aplicación no tiene ninguna observación constructiva o señalamiento sobre el diseño y la funcionalidad, al contrario, señalan que la aplicación es muy intuitiva y puede fácilmente lograr los objetivos que tiene planteados.

6.2. Actividad de observación e interpretación de resultados

Para la actividad de observación por parte del investigador se desarrolló la checklist o lista de verificación mostrada en el Cuadro 6.1 en forma de tabla verificando si se cumplían ciertos aspectos en la investigación. Es muy importante resaltar que esta observación tiene como objetivo identificar la funcionalidad de la aplicación y el nivel de aceptación por parte de los usuarios para la investigación. Esta actividad permitirá



Figura 6.10: Gráfica interpretativa enunciado 10. Fuente: Elaboración propia.

conocer si las hipótesis planteadas se cumplen o no.

6.3. Interpretación de resultados

Después de analizar los resultados obtenidos en el estudio que se realizó a los 30 participantes se puede llegar a la siguiente conclusión sobre las hipótesis planteadas.

Con base en los resultados alcanzados y atendiendo la discusión de los mismos, existe suficiente evidencia para sustentar que mediante la integración de una aplicación móvil como herramienta en los procesos de promoción y difusión sí es posible generar mayor impacto de uno o más procesos del ITSMT.

De esa cuenta se tiene que en el ITSMT implementa tecnologías nuevas para la evolución de sus procesos y generar mayor impacto, pues la integración de los métodos queda debidamente demostrada a través de los procedimientos expuestos en la presente tesis de grado.

Es importante subrayar que la integración de la aplicación se le puede conferir un nivel razonable de confianza, especialmente porque la aplicación y sus tecnologías provienen del empleo de técnicas científicamente aceptadas en el desarrollo de software, por lo que al realizar la investigación donde se empleó técnicas de encuestas estadísticas apli-

Indicador	Si	No
La aplicación pudo instalarse sin problema en todos los dispositivos	X	
Hay usuarios en la muestra que provienen de otra localidad	X	
Los usuarios pueden utilizar por si mismos o de manera intuitiva	X	
la aplicación Presento alguna falla durante la actividad de prueba		X
El proceso de las funciones demora en cargar menos de 5 segundos		X
A los usuarios les interesa la preferencia de otros usuarios		X
La información que dan sobre las preferencias de los usuarios es de utilidad para el ITSMT	X	
Los usuarios comentan sus elecciones estudiantiles en base al uso de la aplicación	X	
Los usuarios refuerzan su decisión posterior al uso de la aplicación	X	

Cuadro 6.1: Checklist del investigador. Fuente: Elaboración propia.

cadadas en situaciones como las que se mencionan en el primer capitulo de la presente investigación y porque adicionalmente fue utilizada una de las técnicas del método cuantitativo, como la encuesta de aceptación. Todos estos elementos permiten admitir que los resultados tengan validez.

Por lo mencionado anteriormente se puede obtener las siguientes conclusiones acerca de las hipótesis.

La promoción y difusión del ITSMT tendrá un mayor alcance geográfico debido a que la aplicación móvil facilita el acceso a más lugares de la región.

Para comprobar si se cumple la hipótesis anterior en la muestra se identificaron varios estudiantes que provienen de diferentes comunidades y se verificaron sus encuestas. La mayoría de todos los usuarios que provienen de otra localidad han comentado que “facilitar el acceso a información desde un dispositivo móvil es muy practico” pues muchas de las ocasiones no se tiene la oportunidad de visitar comunidades o localidades de la región por falta de tiempo o personal en el ITSMT ademas de la dificultad del traslado hasta la ubicación. Solo un usuario comenta que no le conviene el uso de la

aplicación desde su comunidad ya que no hay señal telefónica ni de Internet por lo que la opción de uso queda totalmente descartada. Ahora al saber que la información que se les provee les es de utilidad y cumpliendo la condición de ser estudiantes de localidades diferentes de la región se puede establecer que la primer hipótesis si se cumple.

Los aspirantes estarán mejor preparados en su desempeño académico por que serán orientados para saber que carrera se ajusta a su perfil de acuerdo a sus aptitudes y actitudes.

Para la hipótesis anterior es necesario evaluar el desempeño de todos aquellos aspirantes que fueron seleccionados en la muestra ya matriculados en el ITSMT puesto que se requiere que pongan en marcha su decisión de que estudiar basados en los resultados arrojados por la aplicación, por el momento, debido a que no se cumplen aun los tiempos de que estos participantes de la muestra se integren a una universidad, la hipótesis no puede ser comprobada como valida o no valida, así que el resultado obtenido hasta este punto es indefinido.

Se conocerá la preferencia de los aspirantes en menor tiempo posible en función de las carreras que oferta el ITSMT.

En esta hipótesis el resultado si puede demostrarse su resultado en el tiempo que se realizo la investigación, muchos de los estudiantes en un proceso común de recopilación de preferencias, llenan hojas o encuestas en papel, que después serán conteadas y analizadas por el personal a cargo de los procesos de ferias de promoción y difusión del ITSMT. Con la integración de la aplicación desarrollada, no es necesario realizar un conteo ya que el proceso se mide y se obtienen los reportes estadísticos en forma de gráfica a través de un algoritmo de conteo de resultados y conversión a valores entendibles por la librería que genera las gráficas. Entonces el tiempo que se tardaba en conocer las preferencias de resultados es reducido a unos cuantos segundos ya que se puede conocer cada que se realice una consulta a través del sistema monitor que verifica los datos de los usuarios en tiempo real. La integración de este proceso de software reduce el tiempo que se tardaba en obtener los resultados sobre las preferencias de los estudiantes. Por ello se concluye que la hipótesis si se cumple.

Conclusión

El estudio que se realizo en este capitulo obtuvo resultados muy concretos a excepción de una hipótesis cuya validez no puede ser comprobada debido a que no se coincide

aun el tiempo en el que es realizada la investigación con la temporada que se requiere para realizar un estudio y evaluar este resultado.

Muchas de las técnicas utilizadas para la recopilación y medición de datos no son complicadas, esto con el objetivo de interpretar los resultados de la manera mas comprensible para el lector. Todo con el fin de generar un trabajo de investigación comprensible.

Gracias a los resultados se puede evaluar que el nivel de aceptación de la aplicación y su impacto es muy alto, las pruebas realizadas con la muestra seleccionada ayudaran que se mejore aun mas la aplicación y sus módulos que dedicados unicamente a los procesos de promoción y difusión.

7 Conclusiones

7.1. Conclusión

Gracias al estudio realizado, con las actividades de recolección y análisis de datos se pudo llegar a la conclusión y aceptados de solo dos hipótesis de las 3 planteadas. Debido a la temporada, la segunda hipótesis aun no puede tener un resultado definido y concreto. Por ello un estudio extra basados en algunos trabajos a futuro puede definir si se cumple o no. Obteniendo solo dos resultados favorables se puede decir que la herramienta desarrollada genera un impacto aceptable y que puede mejorar el proceso de promoción y difusión del ITSMT de varias maneras, apoyando a los aspirantes con su elección educativa, recopilando datos y transformándolos en información útil, brindar información confiable y segura a los aspirantes, etc.

El proyecto se fue desarrollando de acuerdo a las etapas contempladas en el cronograma de actividades de la investigación para evitar desfasamientos de tiempo, esto tuvo como resultado que el sistema monitor y la aplicación se terminaran en tiempo y forma, sin contra tiempos. A su vez, el desarrollo en tiempo y forma de la aplicación ayudo a que se pudieran realizar pruebas y estudios sobre su impacto en lugar de su desarrollo, lo que comúnmente es lo que una aplicación móvil busca, generar un gran impacto en los usuarios finales para que aumente su uso y de un beneficio de a ambas partes.

De acuerdo a todos los conocimientos adquiridos durante la carrera de ingeniería en sistemas computacionales en semestres anteriores a la residencia profesional y la experiencia en servicio social se pudo obtener la capacidad suficiente para poder desarrollar un producto funcional y también llevar a cabo la investigación de su impacto. Cabe mencionar que muchas de las tecnologías que se utilizaron en este proyecto no fueron abordadas en las materias cursadas durante la carrera. Por ello, el autodidactismo tuvo que ser la principal motivación para desarrollar esta aplicación e investigar su nivel de impacto y aceptación.

A lo largo del desarrollo del proyecto y la investigación con la herramienta se obtuvieron buenas experiencias tales como trabajar bajo un ambiente agradable ya que se lograron cubrir los requerimientos solicitados por el cliente (ITSMT) con los módulos propuestos para la aplicación y de esta manera realizar una investigación con una herramienta completa evaluando al usuario final (Aspirantes).

En cuanto al producto se puede decir que es la mejor manera de cubrir las necesidades en el departamento de promoción y difusión del ITSMT ya que podemos cubrir múltiples puntos los cuales constaban mucho esfuerzo y recursos al departamento, y ahora con la aplicación tenemos una herramienta que además de proporcionar un beneficio al usuario proporciona un beneficio a la institución ya que brinda información útil para la toma de decisiones a futuro del ITSMT.

Realizar un estudio de las características que debe cumplir una tecnología para generar un impacto conlleva muchas complicaciones que a su vez convierten al ITSMT en una institución de calidad, puesto que implementando tecnologías como las que se describen en este proyecto pueden mejorar muchos de los procesos. Todos los datos nos permiten a su vez tener un acercamiento al cliente o usuario final, interactuar con el, da muchas oportunidades de mejoramiento en muchos procesos, puesto que siempre hay que saber como llegar al pensamiento de los jóvenes y motivarlos a formar parte de una carrera que pueda mejorar la calidad de vida a través de una educación que tiene una estructura muy bien definida como la del ITSMT y que constantemente se actualiza en técnicas de enseñanza y en las tecnologías que aumentan la calidad de los procesos de la institución.

Es muy importante señalar y expresar que el resultado obtenido de la investigación puede ayudar a mejorar los procesos y las tecnologías que los apoyan, realizar investigaciones y mejoras a futuro en la aplicación pueden aumentar el impacto de una herramienta móvil y obtener resultados mas concretos que puedan ofrecer un beneficio aspirante-institución.

7.2. Trabajos Futuros

Queda abierta la posibilidad de aumentar la capacidad de la herramienta en diversos aspectos, esto con el objeto de realizar estudios a futuro e intentar obtener resultados sobre las hipótesis cuyo estado es indefinido o no se sabe si se cumplen o no.

En base a lo anterior se puede establecer que los siguientes puntos pueden ser las próximas actividades o trabajos a futuro para profundizar en esta investigación:

Desarrollar y documentar un modulo que pueda dar seguimiento a los usuarios de nivel aspirante a matriculado.

Estudiar las decisiones de los usuarios en base a los resultados y que queden registrados previamente, para evaluar si la aplicación realmente los ayudo en su elección educativa.

Realizar un trabajo de investigación en base a los módulos implementados y nuevas tecnologías que pueden ser interactuar con mas redes sociales cuyo objetivo seria medir un nuevo nivel de impacto con dichas mejoras.

Bibliografía

- [1] Ahonen, T. T., & Barrett, J. (2002). Services for UMTS: Creating Killer Applications in 3G. West Sussex: John Wiley & Sons.
- [2] Anton, D., & Anton, C. (2006). ISO 9001 Survival Guide. Ashland, OR: AEM Consulting Group.
- [3] Mobile Marketing Association. Recuperado el 27 de septiembre de 2017 en <http://www.mmaspain.com/wp-content/uploads/2015/09/Libro-Blanco-Apps.pdf>
- [4] Cecilia, Z. S. (2006). Proceso de contratación de personal. Ciudad de Mexico: Gestipolis. Obtenido de <https://www.gestipolis.com/proceso-de-contratacion-de-personal/>
- [30] Connolly, T. M., & Begg, C. E. (2005). Sistemas de bases de datos 4ta Edición. México: Pearson Educación.
- [24] Deitel, P. J., & Deitel, H. M. (2008). Como programar en C++. México: Pearson Educación.
- [25] Deitel, P. J., & Deitel, H. M. (2009). Como programar en Java. México: Pearson Educación.
- [5] Dunaway, D. K., & Masters, S. (2001). • CMM-Based Appraisal for Internal Process Improvement (CBA IPI Version 1,2 Method Description). Pittsburgh, PA: Carnegie Mellon University. Obtenido de www.sei.cmu.edu/publications/documents/01.reports/01tr033.html
- [6] Escalante, J. L. (2008). Procedimiento para la contratacion de personal. Ciudad de Mexico: Secretaria de Salud.
- [7] Gargenta, M. (2011). Learning Android. Estados Unidos de America: O´Reilly Media Inc.

- [8] Gasca Mantilla, M. C., Camargo Ariza, L. L., & Medina Delgado, B. (2013). Metodología para el desarrollo de aplicaciones móviles. Bogotá: Universidad Distrital Francisco José de Caldas. Obtenido de <http://revistas.udistrital.edu.co/ojs/index.php/Tecnura/article/view/6972/8646>
- [9] Gironés, J. T. (2012). El gran libro de Android Segunda Edición. México: Alfaomega Grupo Editor S.A. de C.V.
- [28] Google Inc. (01 de 01 de 2014). www.android.com. Obtenido de <https://developer.android.com/studio/intro/index.html?hl=es-419>
- [31] Google Inc. (11 de Mayo de 2011). Firebase. Obtenido de <https://firebase.google.com/products/database/>
- [35] Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2010). Metodología de la Investigación 5ta Edición. Ciudad de México: McGraw Hill Companies Inc.
- [10] IBM Corporation. (Abril de 2012). Desarrollo de aplicaciones nativas, web o híbridas. Estados Unidos de America: IBM Corporation. Obtenido de ftp://ftp.software.ibm.com/la/documents/gb/commons/27754_IBM_WP_Native_Web_or_hybrid_2846853.pdf
- [11] Interactive Advertising Bureau. (2017). ESTUDIO DE CONSUMO DE MEDIOS Y DISPOSITIVOS ENTRE INTERNAUTAS MEXICANOS 2017. Ciudad de México: IAB. Recuperado el 20 de 11 de 2017, de <http://www.iabmexico.com/estudios/consumo-medios-2017/>
- [33] Justinmind. (18 de Febrero de 2018). www.justinmind.com. Obtenido de <https://www.justinmind.com/overview>
- [26] Kochan, S. G. (2012). Programación con Objective-C. Primera Edición. Madrid, España,: Anaya Multimedia.
- [12] Master Magazine. (2017). Master Magazine Inc. Obtenido de <https://www.mastermagazine.info/termino/3874.php>
- [13] Meléndez, J., & Murcia, E. (1 de Enero de 2013). Modulo Web Front-End para el desarrollo de simulación a partir de Weibull, Ji cuadrado y Beta. TFC. Recuperado el 13 de Enero de 2018, de <https://www.ucatolica.edu.co/: repository.ucatolica.edu.co/bitstream/10983/1283/3/Paper.pdf>

- [14] Mobile Marketing Association Spain. (2011). Libro Blanco de las Apps. España: Mobile Marketing Association. Obtenido de <http://www.mmaspain.com/wp-content/uploads/2015/09/Libro-Blanco-Apps.pdf>
- [15] Muñoz Andrade, E. L., Fernández, E. D., Gómez Reynoso, J. M., & Arevalo Mercado, C. A. (2015). AIS Electronic Library. Impacto de aplicaciones para dispositivos móviles: un estudio de aceptación. Puerto Rico. Obtenido de <http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1354&context=amcis2015>
- [28] Nívelate. (2018). Guía UNAM (Versión 2.3.1). [Aplicación Móvil]. Descargado de <https://play.google.com/store/apps/details?id=com.institutofuentes.android.guiaunamvg&hl=es>
- [16] Paredes Velazco, M., Dominguez Mateo, F., & Santa Cruz Valencia, V. P. (2012). Programación multimedia y dispositivos móviles. Ciudad de México: RA-MA Editorial.
- [17] Ponce Mendoza, U., Yanez Moreno, V., & Soto Bernal, R. A. (2014). Propuesta Metodológica para Desarrollo de Aplicaciones Móviles para Dispositivos Android. Villahermosa, Tabasco: ResearchGate. doi:10.13140/2.1.2855.8720
- [18] Pressman, R. S. (2010). Ingeniería de Software: un enfoque práctico. México, D.F.: McGraw-Hill Interamericana Editores, S.A. DE C.V.
- [19] Project Management Institute, Inc. (2013). Guía del PMBOK Quinta Edición. Newton Square, Pensilvania: PMI Publications.
- [20] Rahimian, V., & Ramsin, R. (2008). Designing and agile methodology for mobile software development: a hybrid ethod engineering approach. Marrakech, Morocco: Second International Conference on Research Challenges in Information Science. doi:10.1109/RCIS.2008.4632123
- [27] Shepherd, G. (2008). ASP.NET 3.5. Madrid, España.: Anaya Multimedia.
- [21] Software Engineering Institute. (Octubre de 2000). SCAMPI, V1.0 Standard CM-MI ©Assessment Method for Process Improvement: Method Description. Pittsburgh, PA: Carnegie Mellon University. Obtenido de www.sei.cmu.edu/publications/documents/00.reports/00tr009.html.
- [32] Tecnológico de Monterrey (2018) MiTec Móvil (Version 5.0.13)[Aplicación Móvil]. Descargado de <https://play.google.com/store/apps/details?id=com.itesm.mitecmovil&hl=es>

- [22] Vargas Agudelo, F. A. (26 de Octubre de 2011). la ingeniería del software en el desarrollo de aplicaciones para dispositivos móviles. Medellín, Antioquia, Colombia. Obtenido de <http://ojs.tdea.edu.co/index.php/cuadernoactiva/article/download/61/58/>
- [23] Wolber, D., Abelson, H., Spertus, E., & Looney, L. (2011). App Inventor. Canada: O'Reilly Media Inc.