



UAEM

Universidad Autónoma
del Estado de México

CENTRO UNIVERSITARIO UAEM TEXCOCO

“Aplicación de la Metodología Mobile-D™,
para la creación del prototipo de App
Caso de estudio: Draw-HánzÍ, sustentada en AR.”

TESIS

Para Obtener el Título de:

LICENCIADO EN INFORMÁTICA ADMINISTRATIVA

Presentan:

ULISES MORENO GORGONIO

SANDY TREJO QUINTERO

Directora:

M. EN C. YEDID ERANDINI NIÑO MEMBRILLO

Revisores Adjuntos:

DR. EN C. FARID GARCÍA LAMONT

M. EN C. CARLOS OCELOTL RIVERA VILLA



HUMANISMO

QUE TRANSFORMA

TEXCOCO, EDO. MÉX., OCTUBRE 2017



Universidad Autónoma del Estado de México

Centro Universitario UAEM Texcoco

Texcoco, México a 17 de octubre 2017

Asunto: Etapa de digitalización

M. EN C. ED. VIRIDIANA BANDA ARZATE
SUBDIRECTORA ACADEMICA DEL
CENTRO UNIVERSITARIO UAEM TEXCOCO
PRESENTE.

AT'N: L. en D. Marco Rodrigo López González
RESPONSABLE DEL DEPARTAMENTO DE TITULACION

Con base en las revisiones efectuadas al trabajo escrito titulado "Aplicación de la Metodología Mobile-0™, para la creación del prototipo de App caso de Estudio: Draw - Hainci, sustentada en AIP" para obtener el título de Licenciado en Informática Administrativa presentan los sustentantes Ulises Moreno Horgonio y Sandy Trejo Quintero, con números de cuenta 112499 y 9910799 respectivamente, se concluye que cumple con los requisitos teórico-metodológicos por lo que se le otorga el voto aprobatorio para su sustentación, pudiendo **continuar con la etapa de digitalización** del trabajo escrito.

ATENTAMENTE

Farid García Llamant
NOMBRE Y FIRMA DEL REVISOR

Carlos Ocelot Rivera Villa
NOMBRE Y FIRMA DEL REVISOR

**TITULACION
RECIBIDO**
Por González
Texcoco, Méx., a 17 de 10 del 17

Yedid Frandini Nave Membreño
NOMBRE Y FIRMA DEL DIRECTOR

- c.c.p. Sustentantes: Moreno Horgonio Ulises y Trejo Quintero Sandy
- c.c.p. Director de trabajo escrito: M.en C. Yedid Frandini Nave Membreño
- c.c.p. Titulación.- L. en D. Marco Rodrigo López González.

Centro Universitario UAEM Texcoco
Av. Jardín Zumpango s/n. Fracc. El Tejocote
C.P. 56259 Texcoco, Estado de México.
Tels. (595) 9211216 - 9211247 - 9210368 - 9210493
e-mail: cutex.uaem@gmail.com.



A:

A la Deidad *por darme el tiempo y la paciencia para realizarme profesionalmente.*

A mis padres y hermano *por brindarme su apoyo, los recursos necesarios y su confianza en todo lo imprescindible para cumplir mis objetivos personales y profesionales.*

A mi madre Juana Quintero Robles que ha sido el mayor impulso en toda mi carrera y el pilar principal para la culminación de la misma; sin su apoyo constante y amor incondicional, por hacer de mí una mejor persona a través de sus consejos y enseñanzas.

A mi padre Fco. J. Manuel Trejo Núñez *por estar a mi lado apoyándome, siendo mi fuente inagotable de sabiduría, fortaleza y consejo en todo momento.*

A mi hermano Juan Manuel Trejo Quintero *por ser mi fuente de motivación e inspiración para poder superarme cada día más, por estar siempre presente acompañándome y retándome en cada oportunidad que tiene.*

A mi profesora Yedid Erandini Niño Membrillo, *por brindarme con toda la pasión, responsabilidad y conocimiento, la instrucción indispensable a lo largo de mi carrera y de la construcción de este proyecto de tesis.*

A mis profesores Carlos Ocelotl Rivera Villa y al Dr. Farid García Lamont, *por sus acertadas observaciones, enseñanzas y pláticas interesantes llenas de conocimiento.*

A mi mejor amigo Ulises Moreno Gorgonio, *quien magnánimamente me acompañó a lo largo de la confección de este trabajo de tesis, con quien pase muchas frustraciones, pero también muchos asombrosos momentos de sabiduría y conocimiento descomunal. Me enseñaste que, aunque todo estaba en nuestra contra y pese al pronóstico desfavorable, con esfuerzo necesario y la correcta dedicación lo logramos satisfactoriamente...Muchas Gracias, confié en que llegarás a ser un gran Informático mexicano.*

Al resto de mi familia y amigos *quienes sin esperar nada a cambio compartieron su conocimiento, alegrías, tristezas y a todas aquellas personas especiales que a lo largo de estos casi seis años que de una u otra manera voluntaria o involuntariamente, no solo por*

estar presentes aportando buenas cosas, sino por los grandes lotes de felicidad, las diversas emociones y sentimientos que me han llenado de sabiduría para concluir esta meta.

"Hoy quiero expresar mi gratitud a la vida, pues me ha dado muchas cosas maravillosas, cada día me da un hermoso amanecer y una oportunidad más para hacer realidad todos mis sueños al lado de las personas que más quiero. Prometo dar todo lo mejor de mí y construir junto a mi familia un mejor mañana, por lo que agradezco a Dios por todas las bendiciones y abundancia en mi vida".

ATENTAMENTE

SANDY TREJO QUINTERO

A:

Dios. *Por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.*

Mis padres. *Apolonia Gorgonio Flores y José Francisco Moreno Vilchis, por ser el pilar fundamental en todo lo que soy, porque siempre me apoyaron y creyeron en mí, pero más que nada, por su amor.*

Mi hermano. *Elizur Moreno Gorgonio por ser un gran hermano mayor del cual aprendí grandes cosas y más que nada por sus consejos y apoyo incondicional, a su esposa Nancy Brito Domínguez por motivarme y creer en mí.*

Mis maestros. *Yedid Erandini Niño Membrillo, Carlos Ocelotl Rivera Villa y al Dr. Farid García Lamot, por su gran apoyo y motivación para la culminación de mis estudios profesionales y para la elaboración de esta tesis.*

Mi amiga. *Sandy Trejo Quintero por compartir buenos y malos momentos y por apoyarnos mutuamente en nuestra formación profesional y culminación de esta tesis, pero más que nada, por su gran amistad.*

Todo este trabajo ha sido posible gracias a ellos.

ATENTAMENTE.

ULISES MORENO GORGONIO

ÍNDICE

ABSTRACT	I
RESUMEN	III
I. INTRODUCCIÓN	1
1.1 ANTECEDENTES	2
1.2 PLANTEAMIENTO DEL PROBLEMA	4
1.3 OBJETIVOS	7
1.3.1 Objetivo General	7
1.3.2 Objetivo Especifico	8
1.4 HIPÓTESIS	8
1.5 JUSTIFICACIÓN	9
1.6 ALCANCES Y LIMITACIONES	12
1.6.1 Alcances	12
1.6.2 Limitaciones	13
II. EL ESTADO DEL ARTE	15
2 MARCO TEÓRICO	15
2.1 Metodologías de desarrollo de software	15
2.1.1 Concepto de ágil	17
2.1.1.2 Principios de las metodologías ágiles	18
2.1.1.3 Diferencias entre metodologías ágiles vs metodologías tradicionales	21
2.1.1.4 Escenario general de trabajo de una metodología ágil	22
2.1.1.5 Factores que afectan la agilidad de un proceso de desarrollo de software	24

2.1.1.6	Ventajas del uso de las metodologías ágiles	25
2.1.1.7	Metodologías ágiles para el desarrollo de software móvil	26
2.2	INGENIERÍA DE SOFTWARE MÓVIL	31
2.2.1	Concepto de ingeniería de software	31
2.2.2	El proceso de desarrollo de software	33
2.2.3	Concepto de ingeniería de Software móvil	34
2.3	DEFINICIÓN DE CALIDAD	36
2.3.1	Concepto de calidad	36
2.3.2	Problemáticas para definir la calidad	36
2.3.3	Definición de la calidad de software	37
2.3.3.1	Elementos de la calidad del software	40
2.3.3.2	Modelos de calidad de software	41
2.3.3.3	Factores y características que determinan la calidad del software	42
2.4	DEFINICIÓN DE DISPOSITIVO MÓVIL	43
2.4.1	Concepto de dispositivo móvil	43
2.4.2	Tipos de dispositivos móviles	43
2.4.2.1	Sistemas operativos para dispositivos móviles actuales	44
2.4.2.2	Concepto de App, para dispositivos móviles	45
2.4.2.2.1	Características de Apps	45
2.4.2.2.2	Clasificación de las Apps	46
2.4.3	Tipos de usuarios	50
2.5	REALIDAD AUMENTADA	52
2.5.1	Concepto de realidad aumentada (AR)	52

2.5.2 Características generales de la realidad aumentada (AR)	53
2.5.3 Elementos de la realidad aumentada (AR)	54
2.5.4 Tipos de tecnologías realidad aumentada (AR)	56
2.5.5 Herramientas para desarrollo de realidad aumentada (AR)	57
2.5.5.1 UNITY 5.3.5 f1 Y 5.4.0 f3	57
2.5.5.1.1 Requerimientos de sistema de instalación del motor UNITY	58
2.5.5.2 VUFORIA SDK v6.2x	58
2.5.5.2.1 Arquitectura de Vuforia	59
2.5.5.2.2 Requerimiento de sistema para instalación de Vuforia	61
2.5.5.3 LENGUAJES DE PROGRAMACION E IDE	61
2.5.6 EJEMPLOS DE APPS LINGUISTICAS SUSTENTADAS EN REALIDAD AUMENTADA (AR)	63
<hr/>	
CAPITULO III. CASO DE ESTUDIO: DRAW-HÁNZÍ	73
3. CASO DE ESTUDIO: DRAW -HÁNZÍ	73
3.1 RECOLECCIÓN DE INFORMACIÓN	73
3.1.1 Apps educativas y/ generación de Aprendizaje	73
3.1.1.1 Aprendizaje INMERSIVO	75
3.1.2 El idioma CHINO	76
3.2 DISEÑO DE PROTOTIPO DE APLICACIÓN MÓVIL DRAW-HÁNZÍ, CON BASE, A LAS DIRECTRICES DE LA METODOLOGÍA ÁGIL MOBILE-D™	96
3.2.1 Necesidades previas fundamentales de formación para implementar correctamente Mobile-D™, en un desarrollo de software	96
3.2.2 Metodología ágil Mobile-D™ y sus respectivas fases en el proceso de desarrollo del prototipo de App: Draw - HánzÍ	97

3.2.2.1 FASE DE EXPLORACIÓN	98
3.2.2.1.1 Propuesta inicial del producto	98
3.2.2.1.2 Contrato del desarrollo de una aplicación	100
3.2.2.2 Etapas de la fase de exploración	101
3.2.2.2.1 Establecimiento del proyecto	101
3.2.2.2.2 Establecimiento de las partes interesadas (Stakeholders)	101
3.2.2.2.3 Definición del alcance del proyecto	104
3.2.2.2.4 Lista de verificación en la fase de exploración	116
3.2.2.2 FASE DE INICIACIÓN	117
3.2.2.2.1 Configuración del proyecto	117
3.2.2.3 FASE DE PRODUCCIÓN	130
3.2.2.3.1 Iteración #1 Módulo de Traducción	131
3.2.2.4 FASE DE ESTABILIZACIÓN	139
3.2.2.5 FASE DE PRUEBAS Y CORRECCIÓN DE ERRORES	140
IV. RESULTADOS Y CONCLUSIONES	144
4. CONCLUSIÓN, RESULTADOS Y TRABAJOS FUTUROS	144
4.1. RESULTADOS	144
4.2 TRABAJO FUTURO	148
4.3 CONCLUSIÓN	149
4.3.1 Aportes Personales de los autores	150
V. BIBLIOGRAFÍA	153

ÍNDICE DE FIGURAS

Figura 1. Metodologías ágiles _____	26
Figura 2. El proceso de software y su evolución _____	34
Figura 3. Proceso de determinación de la calidad en el desarrollo de software _____	41
Figura 4. Modelos de Calidad del Software _____	41
Figura 5. Enhanced Data Mobile Device _____	44
Figura 6. Ejemplo de las características de una aplicación móvil _____	46
Figura 7. Ejemplo de aplicación nativa _____	47
Figura 8. Ejemplo de aplicación híbrida _____	48
Figura 9. Ejemplo de aplicación web _____	49
Figura 10. Tipos de Usuarios de Dispositivos Móviles _____	50
Figura 11. Taxonomía de realidad mixta según Milgram y Kishing _____	53
Figura 12. Proceso de la realidad aumentada (AR), con marcador (target) _____	54
Figura 13. Ingredientes básicos de la realidad aumentada, ejemplos de uso en Smartphone y en pc con código BIDI _____	55
Figura 14. Ejemplo de realidad aumentada con marcadores _____	56
Figura 15. Ejemplo de realidad aumentada sin marcadores, videojuego “Pokemon Go” _____	57
Figura 16. Ejemplo de motor de desarrollo UNITY _____	57
Figura 17. Envolvimiento de desarrollo Vuforia _____	59

Figura 18. Diagrama de flujo sobre la arquitectura de datos del SDK Vuforia en una aplicación móvil _____	60
Figura 19. Ejemplo de desarrollo y logo de Android Studio _____	62
Figura 20. Aplicación Móvil FluentU para aprender idiomas _____	63
Figura 21. Podcast de Chinese Pod para aprender el idioma chino _____	64
Figura 22. Aplicación Chairman's Bao para aprender chino _____	65
Figura 23. Aplicación móvil Memrise para aprender idiomas _____	66
Figura 24. Aplicación móvil AnkiDroid para memorizar por medio de tarjetas _____	66
Figura 25. Aplicación móvil Hello Talk para conversar en otros idiomas _____	67
Figura 26. Aplicación móvil Duolingo para aprender idiomas _____	68
Figura 27. Aplicación Móvil Skitter para aprender idioma chino y japonés _____	69
Figura 28. Aplicación móvil para aprender el idioma chino chinese skill _____	69
Figura 29. Aplicación móvil Aipoly Vision _____	70
Figura 30. Aplicación móvil TapTapSee _____	71
Figura 31. Aplicación móvil Phonic Tricksters _____	72
Figura 32. Evolución histórica de los caracteres chinos _____	79
Figura 33. Comparativo de los tipos de caligrafía china _____	81
Figura 34. Ejemplo del sentido de escritura tradicional _____	81
Figura 35. Ejemplo del sentido de escritura simplificada _____	82
Figura 36. Ejemplo de escritura Pinyin en teclado de dispositivo _____	83
Figura 37. Ejemplo de traducción de una palabra en español a chino simplificado en Google Translate _____	83

Figura 38. Ejemplo de traducción de una palabra en español a chino simplificado en Google Translate _____	83
Figura 39. Ejemplo de los cuatro tonos de chino mandarín _____	84
Figura 40. Ejemplo de significado, dependiendo el tono del carácter _____	85
Figura 41. Ejemplo de instrumentos para la escritura caligráfica china _____	85
Figura 42. Ejemplo de escritura caligráfica china _____	86
Figura 43. Ejemplo de caligrafía china sobre “Mǐzìgé” _____	87
Figura 44. Ejemplo de diferentes “Mǐzìgé’s”, utilizados para la enseñanza de la caligrafía china _____	87
Figura 45. Carácter Yǒng o Eiji Happô _____	90
Figura 46. Ejemplo de trazo del primer principio fundamental de secuencia en un carácter chino _____	91
Figura 47. Ejemplo de trazo del segundo principio fundamental de secuencia en un carácter chino _____	91
Figura 48. Ejemplo de trazo del tercer principio fundamental de secuencia en un carácter chino _____	92
Figura 49. Ejemplo de trazo del cuarto principio fundamental de secuencia en un carácter chino _____	92
Figura 50. Ejemplo de trazo de la primera excepción de los principios fundamentales de secuencia en un carácter chino _____	93
Figura 51. Ejemplo de trazo de la segunda excepción de los principios fundamentales de secuencia en un carácter chino _____	94
Figura 52. Ejemplo de trazo de la tercera excepción de los principios fundamentales de secuencia en un carácter chino _____	94
Figura 53. Ejemplo de carácter chino de la palabra Dios _____	95

Figura 54. Diagrama de procesos de la metodología ágil de desarrollo de software para aplicaciones móviles Mobile-D™	96
Figura 55. Carta de petición de desarrollo de App	100
Figura 56. Línea del tiempo del proyecto	104
Figura 57. Definición del ritmo del proyecto del prototipo de App	105
Figura 58. Diagrama de casos de uso del profesor	110
Figura 59. Diagrama de casos de uso del Administrador	111
Figura 60. Diagrama de casos de uso del usuario de la App	112
Figura 61. Calendario de formación	115
Figura 62. Arquitectura grafica del prototipo de App	118
Figura 63. Arquitectura funcional del prototipo de App	118
Figura 64. Pantalla de Registro	128
Figura 65. Pantalla de Login	128
Figura 66. Pantalla de Menú principal	128
Figura 67. Pantalla de Acciones	128
Figura 68. Pantalla de Lecciones y/o conocimiento	128
Figura 69. Ejemplo de Tarjeta identificar de “Ayuda”	133
Figura 70. Ejemplo de base de datos de los Tarjet’s en VUFORIA	134
Figura 71. Ejemplo de trabajo en el motor de desarrollo UNITY	134
Figura 72. Ejemplo de código “Back” utilizado en el motor de desarrollo UNITY	135
Figura 73. Ejemplo de código “Botón” utilizado en el motor de desarrollo UNITY	135
Figura 74. Ejemplo de código “DropDown” utilizado en el motor de desarrollo UNITY	135
Figura 75. Ejemplo de código “FocusMode” utilizado en el motor de desarrollo UNITY	136

Figura 76. Ejemplo de código “LevelManager” utilizado en el motor de desarrollo UNITY	136
Figura 77. Ejemplo de código “SonidoControlador” utilizado en el motor de desarrollo UNITY	136
Figura 78. Ejemplo de simulación en el motor de desarrollo de UNITY	137
Figura 79. Ejemplo de código “Video Player” utilizado en el motor de desarrollo UNITY	137
Figura 80. Ejemplo de creación de contenido para video en programa de diseño Flash	138
Figura 81. Ejemplo de simulación de video en el motor de desarrollo UNITY	138
Figura 82. Ejemplo de prototipo preliminar de la App Draw- HánzÍ	138
Figura 83. Ejemplo de Target final para reconocimiento	139
Figura 84. Ejemplo de fondo de pantalla final para App Draw-HánzÍ	139
Figura 85. Ejemplo de fondo de pantalla para la App Draw-HánzÍ en ejecución	139
Figura 86. Ejemplo de credenciales de alumno	140
Figura 87. Diagrama de metodología Mobile-D TM final.	143
Figura 88. Ejemplo de módulos de manual de desarrollo de apps en México, sustentado en la metodología ágil Mobile-D TM	145
Figura 89. Ejemplo de ejecución de la app Draw-HánzÍ	146
Figura 90. Ejemplo de Trabajo en motor de desarrollo UNITY	146
Figura 91. Ejemplo de trabajo futuro	148
Figura 92. Página de Facebook de la Comunidad Methodology Mobile-D TM en México	149

ÍNDICE DE TABLAS

Tabla 1. Diferencias entre metodologías ágiles vs. Metodologías tradicionales _____	21
Tabla 2. Comparativa entre las características básicas o bases observados en el desarrollo de software móvil _____	22
Tabla 3. Etapas del Proceso de Desarrollo de Software Móvil _____	35
Tabla 4. Factores y características que determinan la calidad del software _____	42
Tabla 5. Tipos de sistemas operativos Móviles _____	44
Tabla 6. Ventajas y desventajas de las aplicaciones nativas _____	47
Tabla 7. Ventajas y desventajas de una aplicación híbrida _____	49
Tabla 8. Ventajas y desventajas de una aplicación web _____	50
Tabla 9. Requisitos de sistema para versión UNITY _____	58
Tabla 10. Requerimiento del sistema Vuforia SDK _____	61
Tabla 11. Lenguajes de programación e IDE _____	61
Tabla 12. Requisitos de sistema para Android Studio _____	62
Tabla 13. Clasificación del índice de potencia de un lenguaje _____	78
Tabla 14. Composición de la empresa Desarrolladora de Software Freelance _____	99
Tabla 15. Composición del cliente _____	99
Tabla 16. Especificaciones del producto _____	101
Tabla 17. Definición de las partes interesadas por parte del cliente _____	102
Tabla 18. Definición de las partes interesadas por parte del SUMTSOFT _____	103
Tabla 19. Estimación inicial de horas (esfuerzo) _____	105
Tabla 20. Identificación y estimación de costos humanos iniciales _____	106

Tabla 21. Definición de recursos materiales _____	106
Tabla 22. Identificación y costos indirectos de hardware _____	107
Tabla 23. Identificación de costos indirectos de software _____	107
Tabla 24. Identificación y estimación de costos generales _____	108
Tabla 25. Identificación y estimación de costos de contingencia _____	108
Tabla 26. Identificación y estimación del costo inicial _____	108
Tabla 27. Requerimientos generales _____	109
Tabla 28. Descripción del caso de uso “Verificar avance del usuario” _____	110
Tabla 29. Descripción del caso de uso “Añadir conocimiento” _____	110
Tabla 30. Descripción del caso de uso “Validar pago del usuario” _____	111
Tabla 31. Descripción del caso de uso “Consultar información del usuario” _____	111
Tabla 32. Descripción del caso de uso “Verificar avance del usuario” _____	112
Tabla 33. Descripción del caso de uso “Realizar pago” _____	112
Tabla 34. Descripción del caso de uso “Validar Email” _____	113
Tabla 35. Descripción del caso de uso “Hacer Login” _____	113
Tabla 36. Descripción del caso de uso “Consultar Secciones” _____	113
Tabla 37. Descripción del caso de uso “Consultar Acciones” _____	114
Tabla 38. Descripción del caso de uso “Consultar información de las Acciones” _____	114
Tabla 39. Descripción del caso de uso “Consultar información de las secciones” _____	114
Tabla 40. Descripción del caso de uso “Consultar historial” _____	115
Tabla 41. Ejemplo de necesidades de información _____	115
Tabla 42. CheckList del plan proyecto en la fase de exploración _____	116
Tabla 43. Recursos físicos del cliente _____	119

Tabla 44. Definición de comunicación del equipo desarrollador con el cliente _____	120
Tabla 45. Ejemplo de plantilla de Historia de Usuario _____	122
Tabla 46. Historia de Usuario de “Diseño” _____	122
Tabla 47. Historia de Usuario de “Estilo” _____	123
Tabla 48. Historia de Usuario de “Utilidad” _____	123
Tabla 49. Historia de Usuario de “Experiencia” _____	123
Tabla 50. Historia de Usuario de “Atractivo” _____	124
Tabla 51. Historia de Usuario de “Apariencia” _____	124
Tabla 52. Historia de Usuario de “Acceso” _____	124
Tabla 53. Historia de Usuario de “Legibilidad” _____	125
Tabla 54. Historia de Usuario de “Disponibilidad” _____	125
Tabla 55. Historia de Usuario de “Escalabilidad” _____	125
Tabla 56. Historia de Usuario de “Lanzamiento” _____	126
Tabla 57. Historia de Usuario de “Adaptabilidad” _____	126
Tabla 58. Historia de Usuario de “Entereza” _____	126
Tabla 59. Historia de Usuario de “Intromisión” _____	127
Tabla 60. Historia de Usuario de “Inmunidad” _____	127
Tabla 61. Planificación de fases _____	129
Tabla 62. Checklist del plan de la fase de inicialización _____	130
Tabla 63. Historia de Usuario del reconocimiento de “Tarjet´s” _____	132
Tabla 64. Historia de Usuario del reconocimiento de “Tarjet´s- Objetos” _____	132
Tabla 65. Historia de Usuario del Reconocimiento de “Tarjet´s- Video” _____	133
Tabla 66. Tarjeta de tarea T01 relacionada con la función F01 _____	133

Tabla 67. Tarjeta de tarea T02 relacionada con la función F01 _____	134
Tabla 68. Tarjeta de tarea T01 relacionada con la función F03 _____	137
Tabla 69. Historia de usuario #004 requisitos de experiencia, terminada. _____	141
Tabla 70. Historia de usuario en funciones F01, terminada. _____	142
Tabla 71. Tarjeta de tarea F01-T02. _____	142

ABSTRACT

The purpose of this thesis project has been to implement the knowledge and skills acquired during the course of the degree in Administrative informatics, at the University Center UAEM Texcoco; likewise, those obtained in a self-taught way; On agile development Methodologies for mobile software (APPS, for the English language acronym) and the manipulation of new technologies in the field of augmented Reality (AR, for the English language initials).

The result obtained, is the theoretical/practical knowledge derived from the creation and execution from the case of study: Draw-Hánzǐ, a prototype of APP supported and executed in the technological field of the AR, whose objective is to become a tool of learning optimal in language teaching. However, for practical purposes of this thesis project, it is limited by focusing initially on the teaching and learning of the Mandarin Chinese language, basically focusing on pronunciation, translation of Chinese characters in real time, sing common images and the correct calligraphic technique of the same characters, in an easy, practical and intuitive way, supported correctly and effectively in a continuous immersive linguistic learning method, called by experience, this same is It is based largely on the cream capacity that most people develop through direct experience; In other words, reaching the aforementioned practice, help to procreate instruction during its execution, achieving the generation of a practical manual "Development of Apps in México", whose intention is to serve as a standardized guide for the Production of mobile applications (APP, for the English language acronym), in the sector of software development freelance, start-up. and/or PyMEs IT, in México, based on the guidelines aptly marked by the methodology *Mobile-DTM*.

At the same time, an attentive invitation is promoted to the general public and to the specialized personnel, to enter to the infinite world of the development of Apps, with the support of the specialized methodologies for this purpose, besides being interested in confirming of form Practical and simple; The infinite possibilities of AR, in the efficient solution of problems and situations presently present. In such a way, as to reinforce and improve the training on the basic concepts necessary in the

manipulation of the AR, in current mobile devices (tablets, smartphones, consoles, SmartWatches, mobile computers, among others). Consequently, this thesis project will serve as a basis for developing many and varied future Apps, in this same and/or different category of execution.

RESUMEN

El propósito del presente proyecto de tesis, ha sido el implementar los conocimientos y habilidades, adquiridas durante el transcurso de la Licenciatura en Informática Administrativa, en el Centro Universitario UAEM Texcoco; así mismo, los obtenidos de manera autodidacta; sobre las metodologías ágiles de desarrollo para software móvil (*Apps*, por su acrónimo en el idioma inglés) y la manipulación de las nuevas tecnologías en el rubro de la realidad aumentada (AR, por sus siglas en el idioma inglés).

El resultado obtenido, es el conocimiento teórico/práctico de la aplicación de la metodología ágil *Mobile-DTM*, un híbrido de muchas técnicas, prácticas y métodos habituales en el desarrollo de software ágil a nivel mundial, instituida mediante la cooperación estrecha de la industria (empresas reconocidas) de TI Finlandesas y los investigadores del VTT Finlandia; lo que hace de *Mobile-DTM* una metodología cercana a las reglas de desarrollo de App comerciales a nivel mundial, convirtiéndola en la metodología perfecta al contexto del desarrollo ágil de Apps en México, misma propuesta que se comprueba satisfactoriamente en el presente trabajo de tesis, derivado de la creación y ejecución a partir del caso de estudio: *Draw-Hánzǐ*, un prototipo de App sustentada y ejecutada en el rubro tecnológico de la AR, cuyo objetivo es volverse una herramienta de aprendizaje óptima en la enseñanza de idiomas. Sin embargo, para fines prácticos del presente proyecto de tesis, se delimita centrándose inicialmente solo, en la enseñanza y aprendizaje del idioma Chino Mandarín, enfocándose básicamente en la pronunciación, traducción de caracteres chinos en tiempo real, empleando imágenes comunes y la correcta técnica caligráfica de los mismos caracteres, de forma fácil, práctica e intuitiva, apoyada correcta y eficazmente en un método de aprendizaje lingüístico inmersivo continuo, denominado por experiencia, este mismo se fundamenta en gran medida, a partir de la capacidad nata, que la mayoría de las personas desarrollan a través de la experiencia directa; en otras palabras, alcanzando de la antes mencionada práctica, la ayuda para procrear instrucción durante su ejecución, logrando la generación de un manual práctico “Manual de desarrollo de Apps en México”, cuya intención es servir de guía estandarizada, para la producción de aplicaciones móviles (*App*, por su acrónimo en

el idioma inglés), en el sector de desarrollo de software *Freelance, Start-up. y/o* PyMEs IT, en México, con base, en las directrices acertadamente marcadas por la metodología *Mobile-DTM*.

Al mismo tiempo se promueve una atenta invitación al público en general y al personal especializado, adentrarse al infinito mundo del desarrollo de Apps, con el soporte de las metodologías especializadas para este fin, además de estar interesado en confirmar de forma práctica y sencilla; las infinitas posibilidades de la AR, en la solución eficiente de problemáticas y situaciones que se presentan actualmente. De tal manera que se refuerce y mejore la formación sobre los conceptos básicos necesarios en la manipulación de la AR, en dispositivos móviles actuales (Tablets, Smartphones, Videoconsolas, SmartWatches, Computadoras Móviles, entre otras).

En consecuencia, el presente proyecto de tesis servirá como base para desarrollar muchas y más variadas Apps futuras, en este mismo y/o diferente rubro de ejecución.

Capítulo I

INTRODUCCIÓN

El presente proyecto de tesis es una investigación científica tecnológica que tiene por objetivo evaluar la efectividad en la aplicación de la Metodología ágil *Mobile-DTM*, al crear una App en los sectores de desarrollo *Freelance*, *Start-up*, PyME IT en México, con y sin experiencia previa en la producción asistida para productos de software móvil.

Donde como punto de partida, para poder brindar la certeza de que *Mobile-DTM*, es la mejor opción para brindar el apoyo al desarrollo de Apps en el contexto mexicano, se presenta de manera satisfactoria, la creación del prototipo de aplicación móvil (*App*, por su acrónimo en el idioma inglés), caso de estudio: *Draw-Hánzǐ*, sustentada en el rubro tecnológico de la realidad aumentada (*AR*, por sus siglas en inglés), cuyo único objetivo es volverse una herramienta de aprendizaje óptima en la enseñanza de idiomas. Se centra inicialmente exclusivamente, en la enseñanza y aprendizaje del idioma Chino Mandarín, básicamente en: la pronunciación, traducción de caracteres chinos en tiempo real, empleando imágenes comunes y la correcta técnica caligráfica de los mismos caracteres de forma fácil y práctica; apoyada correcta y eficazmente en un método de aprendizaje inmersivo lingüístico continuo, denominado por experiencia, este mismo se fundamenta en gran medida, en la capacidad nata que la mayoría de las personas desarrollan, a través de la experiencia directa.

En consecuencia, de lo antes expuesto con el presente trabajo de tesis, se propone una alternativa de solución divulgativa, mediante la generación de una guía práctica denominada “Manual de desarrollo de Apps en México”, para la producción de Apps, en los antes mencionados sectores en México, con base, en la metodología ágil *Mobile-DTM*, cuyo propósito es procrear conocimiento durante su ejecución, es decir, a partir de la experiencia de su instrucción, logrando que tanto el público en general y/o especializado comiencen a generar Apps de alta calidad, con la finalidad de que ellos mismos observen los beneficios que se obtendrán de introducir dicha propuesta en sus ámbitos de desarrollo *Freelance*, *Start-up*, PyME IT en México.

Así mismo, para la correcta interpretación de la metodología ágil *Mobile-DTM*, será necesario utilizar los conceptos de ingeniería de software básicos generales, por lo que se hace

necesario, la consulta de documentación bibliográfica sobre la Metodología, que proviene de años de investigación por parte de los autores de la misma, en especial de sus artículos, sus talleres, sus proyectos, sus simulaciones, entre otras; en diversos lugares y contextos del mundo. La bibliografía fue contrastada con la realidad de nuestro país en relación a la cuestionable situación por la que atraviesan los desarrolladores de productos de software móvil en México.

Esta investigación presenta los siguientes capítulos:

En el capítulo I se presentan los antecedentes, el planteamiento de la investigación, la problemática, los objetivos, la justificación, los alcances y limitaciones de la misma, además de la relevancia de la investigación.

En el capítulo II se abordan los aspectos relacionados al marco teórico/ conceptual y aspectos metodológicos como son: conceptos sobre de las Apps, los dispositivos móviles, los sistemas operativos móviles y la AR, el motor de desarrollo *UNITY*, el SDK (Kit de desarrollo de software) *VUFORIA*, el lenguaje de programación *C#* y *JavaScript*, el IDE (Entorno de desarrollo integrado) Visual Studio 2016 y 2017, los IDE *Android Studio* y *SDK Build Tools*, Java y Java *JDK* (Kit de desarrollo de Java, conjuntamente, lo relacionado a las Apps educativas y el idioma Chino mandarín para la generación de contenido a utilizarse en el prototipo de App *Draw-Hánzì*.

En el capítulo III se abordan todos los aspectos relacionados a la implementación de la Metodología ágil *Mobile-DTM*, durante el proceso de creación del prototipo de App *Draw-Hánzì*.

En capítulo IV se ofrece la discusión e interpretación de los resultados, así mismo, se presenta la conclusión, las recomendaciones y trabajos futuros de esta tesis.

1.1. Antecedentes

Desde los orígenes de la humanidad, el crecimiento y expansión de la población, por consiguiente, de la sociedad a la par; ha originado la aparición de diferentes tecnologías, en la misma medida, así mismo, se ha revolucionado la concepción y la relación entre ellas. Hoy en día estas tecnologías, en su mayoría, han dotado de herramientas a la sociedad en

sí misma, para afrontar eficazmente una variedad de problemas que hasta entonces, no tenían solución alguna, con los medios existentes en ese preciso momento.

He aquí que la creación de aplicaciones móviles (*Apps*, por su acrónimo en el idioma inglés), producto de la manipulación de la tecnología, han modificado la forma de interactuar entre las personas mundialmente. Hoy por hoy, se puede leer un periódico, consultar la cartelera del cine, revisar el estado de una cuenta bancaria, editar fotos al instante, navegar en redes sociales o simplemente jugar un videojuego, a manera de pasar el tiempo. No importa el lugar en el que se esté, mientras se tenga acceso a un dispositivo móvil con o sin Internet, las Apps siempre estarán al alcance de todo el mundo, independientemente de la edad, estrato social y/o género.

El desarrollo de Apps, hasta el momento ha surgido principalmente por la necesidad de las personas de encontrar soluciones a problemas comunes, de una forma innovadoramente incluyente; lo que determina su popularidad en sí. No obstante, su desarrollo se vuelve un sector de crucial de oportunidades importante como lo demuestra el hecho de que actualmente el país ocupa en segundo lugar en América Latina en la creación de Apps, solo después de Brasil y su crecimiento está siendo aprovechado por únicamente por 10 empresas Start-ups y compañías, entre otras, lo que según la Asociación Mexicana de la Industria de Tecnologías de la Información (AMITI), facturo para el 2014, 400 millones de dólares, el doble de lo que se facturo en el 2013¹. Contexto que debe ser aprovechado en México, específicamente el Estado de México, que a la fecha no presenta información en su tasa de adopción de Apps con calidad por parte de la ciudadanía². Dicho de otro modo, concebir un proceso de creación de un producto de software móvil, cualquiera que este sea; sin el uso de una metodología de desarrollo de software (conocida en el argot informático como Ingeniería de Software misma, que hace más eficaz la producción, poniendo un enfoque estructurado y cierto énfasis al entorno de desarrollo, en el cual se plantea la gestión de un producto de software); no resulta ser lo más recomendable, en ningún contexto imaginado.

Si consideramos que en las últimas dos décadas o posiblemente un poco menos, la creación de productos de software móvil, es decir, Apps, se ha manifestado con mayor frecuencia;

¹ Artículo publicado en el periódico EL FINANCIERO , “10 firmas que lideran el desarrollo de las aplicaciones móviles en México”, por Carolina Ruiz el 30-12-2014.

² Artículo publicado en la revista FORBES MÉXICO, “Reprueban en adopción y efectividad apps gubernamentales”, por Mauricio Hernández Armenta el 31-05-2017.

permeando así la cotidianidad de la vida, en todos los aspectos y sentidos, dentro de la sociedad. Resulta muy extraño entonces, que en la gran mayoría de estos desarrollos, se observen y contemple desde el punto de vista “desarrollador”; la falta del respaldo necesario durante su proceso de desarrollo e implícitamente encontrar publicadas muchas Apps de baja calidad, donde se podría llegar a afirmar que las expectativas (requerimientos iniciales y/o finales) del consumidor, no fueron satisfechas exitosamente, en consecuencia, es de vital importancia introducir a los equipos de desarrollo, el uso de metodologías de desarrollo de software móvil, dentro de las empresas desarrolladoras de software en México (*Freelance, Start-up, PyME IT* y demás), a manera de generar competitividad y calidad en los productos de software que desarrollan.

1.2. Planteamiento de Problema

Para muchas organizaciones en el mundo, el desarrollo de Apps es una forma inteligente de invertir tiempo y dinero; ya que estas, pueden ayudar significativamente a impulsar la productividad en varios aspectos y niveles dentro de las mismas estructuras. Por lo que se observa, cada día más, la tendencia en el uso de las metodologías ágiles para el efectivo desarrollo de dichas Apps; debido al aumento en la demanda, su desarrollo y producción, tiende hacia el enfoque puntual de la explotación; en las destacadas particularidades de las diferentes metodologías de trabajo y/o desarrollo, tanto en lo tradicional como en la agilidad, dependiendo del proyecto del que se tratase, respectivamente.

Sin embargo, las metodologías ágiles están ganando mucho más terreno, y más rápidamente, debido a que muchas empresas desarrolladoras de software en el mundo apuestan por obtener lo mejor, del uso conjunto de estas, con las metodologías predictivas en el mismo desarrollo y/o proyecto. En donde lo ágil gestiona muy bien, la duda, el riesgo y la respuesta rápida; mientras que las metodologías predictivas gestionan mejor la incertidumbre en los proyectos. Un buen ejemplo de estas es la Metodología ágil *Mobile-DTM*, producto de la mezcla de las metodologías de desarrollo de software ágiles, programación extrema (*XP*, por sus siglas en inglés), para las prácticas de desarrollo; metodologías *crystal clear* (*CC*, por sus siglas en inglés), para la escalabilidad y proceso unificado racional (*RUP*, por sus siglas en inglés), como base del diseño del ciclo de vida.

Hay que tomar en cuenta que la asequibilidad de las Apps, a través del mundo entero ha contribuido en el auge de los dispositivos móviles por sobremanera, pero específicamente en México, ha tenido, un gran impacto también. Es aquí donde al dar una mirada alrededor, se puede percibir, cómo las Apps están ingresando a la vida cotidiana de las personas, a modo de tendencia irreversible y con una rapidez inaudita, gracias a la facilidad y experiencia de acceso que proporcionan, lo que lleva a adentrarse cada minuto más en la generación APPLICANDONOS³ en todo y para todo. Esta situación ha cambiado el interés, hacia las Apps, en México; primero por parte del punto de vista del cliente, usuario y/o consumidor, seguido de los desarrolladores Freelance, *Start-ups* y *PyMEs IT* de software mexicanas, que están mostrando hoy en día mayor interés, en producirlas, para “N” situaciones y contextos. Aunque con la particularidad estricta de ser creadas, en el menor tiempo posible, omitiendo en la mayoría de los casos, las características de alta calidad y funcionalidad, que el mercado demanda implícitamente.

He aquí, que se nota con mayor fuerza, como la permeabilidad que tienen las Apps se lleva a cabo con relativa facilidad. No obstante, su desarrollo no es y probablemente nunca será una tarea para nada fácil y mucho menos sencilla de realizar, aun cuando así lo parezca. Aunque, lo crucial de la tarea de crear Apps en México se torna muy compleja y preocupante, debido a distintos factores, creencias y/o motivaciones actuales. Es decir, la problemática radica cuando se analizan las miles de Apps mexicanas, disponibles en las *Apps Stores* (servicio para un OS específico en los dispositivos móviles, donde se encuentran alojadas miles y/o millones de Apps, listas para su descarga e instalación de forma gratuita o a cambio de un pago monetario) y arrojan como resultado, que en su inmensa mayoría están creadas como *hobby* (actividad u ocupación que se realiza meramente por placer durante el tiempo libre) o profesionalismo deficiente, es preciso remarcar lo crucial de este punto; aquí es donde el proceso de desarrollo de una App de alta calidad en México, se vuelve preocupante y en algunos casos alarmantes, debido a que si bien si son desarrollos, hechos en el menor tiempo posible; durante su proceso, se hace caso omiso a un adecuado fundamento sostenible de: las reglas de procesos, buenas prácticas, políticas de negocio, se actúa sin la claridad necesaria del modelo de negocio y la óptima resolución del problema a atacar. Por el contrario, de tomarse de forma eficiente

³ Término que se utiliza en el presente proyecto de tesis, para indicar que actualmente las personas realizan tareas y/o procesos de todo tipo, en su vida cotidiana conjuntamente a través de una aplicación móvil y un dispositivo móvil.

conlleva a una debida sistematización de tareas para una creación efectiva, lo que implica a su vez, la eficaz publicación de una App con alta calidad y la constante especialización de los desarrolladores tácitamente.

Las causas que originan esta problemática en México pueden ser múltiples y variadas. Sin embargo, para usos prácticos de la presente investigación científica tecnológica, se determina adherirse a la siguiente: en México la mayoría de los desarrolladores de software, sin distinción de constitución (*Freelance*, dentro de una *Start-up* y/o *PyME IT*), requieren de la capacitación, investigación, práctica y conocimiento, adecuado sobre las nuevas tendencias de desarrollo para software móvil en el mundo. Igualmente, dicho desconocimiento teórico y/o práctico, ha acarreado un atraso en el entorno de desarrollo Apps con alta calidad en México. De no atenderse la antes mencionada problemática, se generarán múltiples y considerables consecuencias a corto y mediano plazo, dentro de las que se ha detectado como principal, la publicación de Apps con muy baja calidad y funcionalidad, aun cuando algunas de ellas cuentan con las últimas tendencias de ejecución, saturando el mercado, causando un hartazgo en el usuario, cliente y/o consumidor y originando una pésima experiencia que en algunos casos será irreversible o muy difícil de revertir en el mediano y largo plazo; lo que implícitamente mermara su desarrollo y suscitará frustración en los desarrolladores.

Como profesionales en la materia, se tiene la certeza, sobre la importancia y la gran ayuda de generar recursos prácticos. Siendo el caso del presente proyecto de tesis, en el que se propone como la mejor opción de desarrollo para una App, cualquiera que esta sea: la metodología ágil *Mobile-DTM*, como pedestal inicial, al adentrarse al mundo del desarrollo de Apps en México, así como en su correcta ejecución práctica.

Con base en lo anterior y a la experiencia de los autores del presente proyecto de tesis, surgen las siguientes preguntas de investigación:

1. ¿Cuáles son los beneficios que se obtendrán al aplicar la metodología ágil *Mobile-DTM*, en la creación de una App?
2. ¿Mejorará significativamente, la calidad (cumplimiento de requerimientos) de una App, al incorporar la metodología ágil *Mobile-DTM* en el proceso de su creación?

3. ¿Existe relación alguna entre el desconocimiento y desuso de la Metodología ágil *Mobile-DTM*, en los desarrolladores de Apps en México?

1.3. Objetivos

A pesar de los múltiples estudios cursados en la licenciatura, el problema de desarrollar un producto de software, cualquiera, es muy complejo de abordar. Desarrollar Apps funcionales e integrar al mismo tiempo, las óptimas soluciones en los procesos de su desarrollo, que mejoren el desempeño de los creadores (desarrolladores), es el tema de la presente investigación científica tecnológica. Es en este sentido se presentan los siguientes objetivos:

1.3.1. Objetivo General

El principal objetivo de la presente investigación científica tecnológica es demostrar que el uso de la Metodología ágil *Mobile-DTM*, como pedestal inicial en la generación de productos de software móvil, en la creación de cualquier App, con un equipo de desarrollo pequeño y con mínima experiencia en el desarrollo de las mismas, es lo más apropiado, ya que permite obtener una excelente calidad en el producto de software móvil (ejecución de requerimientos), la optimización de los recursos (conocimientos / recursos humanos, económicos y tecnológicos), reduciendo la inversión de tiempo – esfuerzo - dinero, garantizando la correcta satisfacción del cliente, usuario y/o consumidor final, a la vez de generar una competencia continua correcta, en el desarrollador.

1.3.2. Objetivos Específicos

1. Resaltar la importancia de identificar los conceptos básicos para la aplicación de la ingeniería de software, en el desarrollo de Apps, mismos que serán la base teórica del presente proyecto de tesis.
2. Realizar una aproximación empírica a la creación de productos de software móvil, con un prototipo de App, caso de estudio: *Draw-HánzÍ*; utilizando las bases y directrices que marca la metodología ágil *Mobile-DTM*.
3. Mejorar la especialización y conocimiento sobre la manipulación de entornos de desarrollo de la *AR*, *UNITY* y *VUFORIA*, asimismo lenguajes de programación *C#* y *JavaScript*, para operarse en los actuales dispositivos móviles, a través del antes mencionado prototipo de App.
4. Proporcionar una nueva comprensión del desarrollo de App en México, siguiendo las buenas prácticas del mismo, mediante la creación de un manual práctico, que servirá de guía estandarizada para la creación, generación y/o producción de App en México, tomando como base la metodología ágil *Mobile-DTM*.

1.4. Hipótesis

HIPÓTESIS 1: La implementación de la metodología ágil *Mobile-DTM* en la creación de una App mexicana, específicamente comprobado con el prototipo "*Draw-HánzÍ*", aporta múltiples beneficios, dando certeza de la correcta comprensión del proceso de desarrollo del producto de software móvil, cualquiera que este sea en México.

HIPÓTESIS 2: Las directrices y características que emplea, la metodología ágil *Mobile-DTM*, contribuirán a producir implícitamente productos de software móvil Apps, con mayor y alta calidad, conforme la experiencia y conocimiento en su aplicación crezca.

HIPÓTESIS 3: Actualmente existe una relación directa entre el desconocimiento y desuso de la Metodología ágil *Mobile-DTM*, en los desarrolladores de software Móvil en México, mismo que se ve plasmado, cuando los desarrolladores; generan productos de baja calidad, causando insatisfacción de los clientes y no obteniendo la correcta ejecución de los requerimientos del mismo.

Las hipótesis enunciadas anteriormente, tendrán una explicación más detallada, en el presente proyecto de tesis, en lo sucesivo.

1.5. Justificación

Cuando se es novato en el mundo del desarrollo de Apps, en México, hay que tener en cuenta algunos puntos importantes, pero la mayoría de las veces, por la falta de experiencia en la creación de las mismas, no se aprecia lo crucial de su existencia. Sin embargo, es significativa la importancia de su existencia; debido a la considerable inversión de tiempo, que la mayoría de los usuarios y/o consumidores, ejercen en el uso (cotidianidad) de Apps, por sobre el manejo de un navegador, al momento de emplear un dispositivo móvil cualquiera y con la finalidad específica, para lo que fueron creadas.

Según la información obtenida, mediante la breve encuesta, “Uso de las Aplicaciones Móviles (Apps) en México”, lanzada mediante *Google Forms* y distribuida a través del Facebook, en el perfil personal de los autores del presente proyecto de tesis; misma que arrojó como resultado, que el 50 % de los usuarios de dispositivos móviles, tienen más de 15 Apps⁴ instaladas (en un promedio de tres dispositivos por usuario), en los mismos; por consiguiente con ellas realizan una infinidad de actividades como son: entretenimiento, uso de redes sociales, comunicación, lectura, transporte, búsqueda de ofertas de empleo, fotografía, audio, uso como herramienta para realizar su trabajo, solo por mencionar algunas. Las Apps actualmente forman parte de nuestros micros momentos diarios y

⁴ Resultados arrojados por la “Uso de las aplicaciones móviles (Apps) en México”, [Encuesta de Google, lanzada por Facebook], (Moreno Gorgonio & Trejo Quintero, 2017), (Véase Anexo I y/o acceda a <https://goo.gl/forms/JAxVfdULaCJ8OGXi2>).

el 37.5 % de las personas dedican un tercio de su día, siendo un aproximado de entre 5 a 8⁵ horas diarias en utilizarlas, para las actividades antes mencionadas. Las Apps están desempeñando funciones claves, siendo la más importante, tomar decisiones racionales acertadas, mismas que van desde la fidelidad hacia una marca, comunicación crucial y/o acudir físicamente a un lugar para realizar una actividad inmediata (Moreno Gorgonio & Trejo Quintero, 2017) véase anexo I.

Por todo lo anterior, se considera trascendental el uso de un modelo, método y/o metodología, que se adapte a las necesidades y características particulares del desarrollo de Apps en México y obtener así, productos de software móvil con mayor calidad; poniéndolas a disposición de los miles de usuarios y consumidores, que lo demandan día con día, garantizándoles una satisfacción total, al utilizarlas correctamente, ofreciéndoles practicidad simultáneamente, para realizar las antes mencionadas actividades en su día a día.

Debido a los importantes y significativos beneficios que ofrece la metodología ágil *Mobile-DTM*, como son: el minimizar riesgos durante el proceso, el aumento significativo en la productividad de la empresa y/o desarrollador, mayor comunicación entre los mismos, al ofrecer exactamente, lo que el cliente necesite en el menor tiempo posible, generando las entregas justas del producto, garantizando así la funcionalidad en sí misma y por consiguiente la satisfacción total del mismo; considerando y coincidiendo a la vez con los casos de éxito que lo demuestran, en lo sucesivo, es que se toma la decisión de ampliar el conocimiento y difusión de la antes mencionada metodología ágil *Mobile-DTM*, en México, teniendo como base el presente trabajo de investigación científica tecnológica.

En Latinoamérica ya también es una inclinación inminente la implementación de la metodología ágil *Mobile-DTM*, como lo muestran, Meneses Sánchez & Laveriano Meca (2016), que estudiaron el “Prototipo de aplicación móvil utilizando la metodología *Mobile-DTM*, para la verificación de la formalidad en el servicio de taxi metropolitano en la ciudad de Lima”, obteniendo el siguiente resultado: al aplicar la metodología de desarrollo *Mobile-DTM*, para el desarrollo de la App, este mismo uso permitió agilizar y organizar el desarrollo

⁵ Resultados arrojados por la “Uso de las aplicaciones móviles (Apps) en México”, [Encuesta de Google, lanzada por Facebook], (Moreno Gorgonio & Trejo Quintero, 2017), (Véase Anexo I y/o acceda a <https://goo.gl/forms/JAxVfdULaCJ8OGXi2>).

del proyecto. A su vez también, Chicaiza Angamarca & Morales Caiza (2016), que estudiaron la “Implementación de un sistema domótico de videovigilancia, aplicando la metodología *Mobile-DTM*, sobre la plataforma Android, para el control de la seguridad de las salas docentes de la unidad académica C.I.Y.A- la matriz Universidad Técnica de Cotopaxi en el periodo académico Noviembre 2014- Agosto 2015”, obteniendo el siguiente resultado: las ventajas que se obtuvieron sobre la metodología *Mobile-DTM*, es que no se limita a una sola estructura como metodología ágil de desarrollo, ya que cuenta con una gran variedad de alternativas para este tipo de App, todos sus artefactos se utilizan, acorde a las necesidades del desarrollador de la App, del lugar donde se lo aplique y los requisitos del cliente, obteniendo un producto de calidad y en corto tiempo. Así mismo Guachalla Blanco (2015), estudio la “Plataforma de geolocalización de personas mediante dispositivo Móvil” y obtuvo el siguiente resultado: mediante el uso de la metodología *Mobile-DTM* y SOA al crear la *App Web*, cuyo desarrollo basado en pruebas permitió detectar y corregir errores en una temprana etapa del desarrollo, lo que permite, no arrastrar errores en otras, disminuyendo el impacto de la corrección del error, permitiendo que el software móvil tenga alta calidad.

En México, igualmente se muestra una pequeña inclinación inminente para la implementación de la metodología ágil *Mobile-DTM*, como lo muestran, Florencia Barrera Ángel Fabián, Martínez Sánchez Víctor Manuel y Ramírez Domínguez Luis Mario, que presentaron “Sistema para dispositivos móviles usando cómputo ubicuo para servicios en la Escuela Superior de Cómputo”, obteniendo el siguiente resultado: al utilizar la metodología *Mobile-DTM*, permite en un periodo corto realizar el proceso de desarrollo de software y por ser para el cómputo móvil cuenta con las etapas necesarias para este proyecto. A su vez también, Lara Trejo Verónica y Tapia Caballero Luis Alberto, que estudiaron el “Anteproyecto sobre los tópicos avanzados de bases de datos distribuida”, obteniendo como resultado, que con la aplicación de la metodología *Mobile-DTM*, la empresa Mobinex, en conjunto con su plataforma Smartface permite la publicación rápida de las Apps que combina teléfono, la funcionalidad, la red, el contenido, las características y la interfaz de usuario rica.

Así mismo a nivel mundial se ha utilizado en diversos proyectos en países como Finlandia, Noruega, Italia, Corea del Norte, Estados Unidos, por mencionar algunos desde el año 2004, hasta la fecha con algunos puntos de fluctuación considerables.

Concretamente en este punto, con la investigación científica tecnológica e implementación de la metodología ágil *Mobile-DTM*, en el presente proyecto de tesis, durante la creación del prototipo de App: *Draw-HánzÍ*, se ofrece una gran oportunidad, de trabajar en estrecha colaboración con las tendencias de desarrollo de Apps más actuales, así como los nuevos *gadgets* de ejecución del propio conocimiento. Además de aportar un régimen flexible para poder seguirla, debido a que cuenta con un enfoque de trabajo en equipo que facilita, la división de las contribuciones al proyecto efectivamente. Posibilita ver el desarrollo desde el punto de vista del cliente (permitiendo que el equipo desarrollador, funja como agente revisor del proyecto, así como cliente, otorgando su atinada opinión, en cuanto a las futuras modificaciones ideales para el prototipo de App).

El presente proyecto de tesis proporcionara datos y experiencia a los autores del mismo, que serán útiles para desarrollar y lanzar, un sinfín de Apps con o sin el uso de la AR, así como de otras tecnologías innovadoras, posteriormente. Además de que los resultados obtenidos de la misma investigación beneficiaran al público en general y/o experto, al adentrarse al vasto mundo de desarrollo de las Apps, sustentadas en AR a nivel mundial.

1.6. Alcances y limitaciones

1.6.1. Alcances

El presente proyecto de tesis tiene como alcance el explorar el uso de una metodología ágil *Mobile-DTM*, en el ámbito práctico y/o laboral del desarrollo de software móvil en México, sin importar su constitución, ya sea desarrollador *freelance*, así como dentro de una *Start-up* y/o *PyMEs IT* desarrolladoras de software mexicanas.

Los aspectos específicos que comprende esta investigación científica tecnológica están referidos a los beneficios, al conocimiento y difusión de la metodología ágil *Mobile-DTM*, dentro de las cuales se abarca, muy por encima, los temas de AR, así como Apps educativas generadoras de conocimiento. Con la pretensión de otorgar al lector el referente teórico mínimo básico argumentativo necesario para comprender la antes mencionada metodología, por lo que se omite el uso y conceptualización de algunos temas como lo son: gestión de procesos de negocio, gestión de requisitos, tipos de desarrollo de software, tipos

de viabilidad, interfaz de programación de aplicaciones, por mencionar algunas. Así mismo, en relación a la App, los idiomas de disponibilidad, características del dispositivo (cámara, acelerómetro, GPS), notificaciones push, formas de pago, modelos de monetización, publicidad de distribución, recolección de datos del usuario en paralelo, venta de datos a tercero, entre otras.

Los aspectos relacionados con la enseñanza sobre el idioma chino se analizarán concretamente con fines de generación de contenido, única y exclusivamente como ejemplo, en ningún momento se pretende la modificación o cambio del sistema de enseñanza de ningún idioma en forma alguna.

No se incluirá el costo (económico) beneficio de la implementación de la Metodología *Mobile-DTM*, dentro de una *Start-up* y/o *PyMEs IT* desarrolladoras de software en México, así como en el sector *Freelance*, debido a que no se cuenta con parámetros previos de comparación.

Se reconoce la viabilidad que ofrece el presente proyecto de investigación científica tecnológica, debido a que se cuenta con los medios económicos básicos, equipamiento material primario y de conocimientos elementales previos para llevar a cabo con éxito, el desarrollo del prototipo de App caso de estudio: *Draw-Hánzǐ*, única y exclusivamente hasta las primeras cuatro fases de su desarrollo, tomando en cuenta que la mayoría de las herramientas que se pretenden utilizar, se buscará sean bajo la licencia *Open Source*.

1.6.2. Limitaciones

Las presentes limitaciones restringirán el presente proyecto de tesis:

1) *Falta de instrumentos disponibles para mediciones técnicas*

Por la característica técnica de la presente investigación científica tecnológica algunos aspectos como la medición de la calidad del prototipo de App: *Draw-Hánzǐ* en todos los aspectos (requerimientos, funcionalidad, usabilidad, portabilidad, diseño, entre otras). Requieren de normas *ISO (International Organization for Standardization)* y otros estándares de certificación de calidad, todos éstos de paga y que conjuntamente tienen que ser aplicados por personal calificado y/o certificado en las mismas. Del mismo modo, se requiere de una App completa a evaluar, con los

cuales no se cuentan; esto hace que los datos que se proporcionan en esta investigación sean datos aproximados y no datos exactamente medidos.

2) *Características de la muestra*

Dentro de la población usuaria de las metodologías ágiles de desarrollo para Apps y/o software móvil, en cualquier estrato constitucional “*Freelance y/o Empresa*”, existen varios puestos, como lo son: analista, arquitecto de software, arquitecto de sistema, desarrollador de software, visionario, jefe de desarrolladores, jefe de producto, diseñador gráfico, tester, gerente del proyecto, administrador de cuentas, administrador de sistemas, administrador de código, capacitador, consultor, cliente y usuario de App. Por lo que no se puede obtener una muestra homogénea, debido a que no todas estas personas, independientemente del puesto que desempeñen, cuentan con los conocimientos y la experiencia, suficiente sobre la implementación total de la metodología ágil *Mobile-DTM* y/o metodología ágil diferente, para efectuar una encuesta muy detallada y minuciosa.

3) *Sesgo del sujeto*

Las respuestas que se obtendrán en una única encuesta dependerán del grado de conocimiento que se tenga acerca de la implementación de metodologías ágiles de desarrollo para Apps en México, y en específico de la metodología *Mobile-DTM*, o temas afines, por tratarse de temas expertos, los únicos que cuentan con la información adecuada son los especialistas en el uso de estas herramientas de desarrollo y/o ingenieros en requerimientos e ingeniería de software.

4) *Acceso al sujeto*

Restringido a la disposición de tiempo por parte de los desarrolladores y/o expertos de acuerdo a sus ocupaciones laborales; así como en brindar información que puede ser considerada vital por parte de la empresa desarrolladora y/o cliente, en consecuencia, de sus políticas propias.

Capítulo II

EL ESTADO DEL ARTE

2. MARCO TEÓRICO

2.1. Metodologías de desarrollo de software.

A través del tiempo, se confirma, que el desarrollo de software siempre ha sido una tarea complicada. Prueba de esto, es la creación de las diversas propuestas metodológicas de desarrollo de software, que incurren de forma más efectiva en todas o en algunas de las distintas fases de desarrollo, dependiendo del mismo proyecto.

Las metodologías para el desarrollo de software, son un modo sistemático de realizar, gestionar y/o administrar un proyecto, con la única finalidad de llevarlo a cabo con altas probabilidades, al éxito. Esta manera de organización, indica cómo se dividirá efectivamente un proyecto en módulos, en otras palabras, en etapas, acciones y/o tareas que corresponden a cada una de ellas, revela las entradas y salidas para cada una de estas, pero sobre todo normaliza o estandariza los procesos a seguir, implementar y mantener al realizar un producto de software, desde que surge la necesidad del producto, hasta que se ve realizado el objetivo por el cual se está creando.

Por una parte, están las metodologías tradicionales, que se basan especialmente en el control excesivo del proceso, estableciendo rigurosas actividades que involucran las herramientas y las prácticas que se usaran en las mismas, así como una específica y constante documentación, del proceso paso a paso.

Estas metodologías han sido efectivas y necesarias en un gran número de proyectos, tanto actuales, como unos un poco más antiguos. Sin embargo, también han presentado problemas en muchos otros. Como resultado de esto, han surgido nuevas metodologías de desarrollo que se centran más en el factor humano y en el producto de software final, que en la documentación.

Ahora bien, del mismo modo como se ha mencionado lo anterior, se puede abordar el desarrollo de un producto de software móvil, es decir, de una App con alta calidad, que ayude a la comunicación entre las personas y/o a la consecución de una tarea específica;

mediante el uso de una metodología ágil de desarrollo, es un reto de investigación actualmente, muy importante; misma para la que son variados los autores que se han centrado en su importancia. Siendo algunos de ellos, los siguientes:

En el año 1999 en “Un proceso ligero, de bajo riesgo, flexible, predecible, científico y divertido de desarrollar software”, Kent Beck uno de los más destacados autores en los procesos ágiles de desarrollo de software, creador de la *Extreme Programming (XP)*, una metodología centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software. Quien incansablemente se ha dedicado a promover el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, proponiendo una comunicación fluida entre todos los participantes, así como la simplicidad en las soluciones implementadas, y el coraje para enfrentar los cambios. Ha definido especialmente la forma adecuada de sobre llevar los proyectos con requisitos imprecisos, muy cambiantes y donde existe un alto riesgo técnico.

En el año 2000 en “Una metodología ágil que requiere de gran disciplina”, Alistair Cockburn, otro de los más destacados autores en los procesos ágiles de desarrollo de software, creador de la *Crystal Clear Methodology (CC)*, misma que no es en sí, una sola metodología de desarrollo de software, sino más bien una familia de metodologías con un código genético en común, debido a que se subdivide en varios tipos de metodologías en función a la cantidad de personas que vayan a estar trabajando dentro de un proyecto en particular. *Crystal Clear Methodology* hace más énfasis en la menor documentación y en más versiones del producto que corran y puedan ser probadas. Cockburn afirma que los diferentes tipos de proyectos requieren diferentes tipos de metodologías, viéndolo desde dos ejes principales: el número de personas en el proyecto y las consecuencias de los errores.

En consonancia en 2005, lo ha expresado también, Martin Fowler, en su ensayo “La nueva metodología”, probablemente el cambio más notable al pensamiento de procesos de software en los últimos años ha sido la aparición de la palabra “ágil”. Hablamos de los métodos ágiles de desarrollo de software, hablamos de la manera de introducir la agilidad en un equipo de desarrollo o de la manera de resistir la tormenta inminente de agilistas decididos a cambiar las prácticas bien establecidas.

De manera semejante, el presente proyecto de tesis está diseñado para determinar que las metodologías ágiles de desarrollo de software móvil, específicamente la metodología ágil *Mobile-DTM*, permiten adaptar la forma de trabajo a las condiciones específicas del proyecto y su desarrollo en México, tomando en cuenta las circunstancias específicas del entorno, reduciendo de forma eficaz los costos (esfuerzo-tiempo-dinero) e incrementado la productividad de las empresas mexicanas desarrolladoras de software, permitiendo tanto a los clientes, usuarios y/o consumidores de Apps, como a las mismas empresas desarrolladoras rentabilizar sus inversiones, en cualquier tipo de proyecto de desarrollo más rápidamente; gracias a la realización de entregas tempranas, el cliente tendrá rápido acceso a aquellas funcionalidades que en verdad aportan, valor acelerado en el retorno de la inversión, para ambas partes.

2.1.1. Concepto de ágil.

ÁGIL significa que se mueve con soltura y rapidez o prontitud (Real Academia Española, 2014)

Sin embargo, desde el punto de vista del desarrollo de software, el término ágil está intentando ofrecer una respuesta rápida a la ansiosa comunidad empresarial, que exige, que las metodologías, métodos y/o técnicas sean cada día, más ligeras con mayor soltura, conjuntamente con procesos de entrega más rápidos, pero sin perder la calidad implícita que otorga su uso y de una reducción justa de documentación.

En este sentido las peculiaridades principales de la definición de agilidad para el desarrollo de software son las siguientes:

- I. Agilidad es la habilidad para crear y responder a cambios de acuerdo a los beneficios en un ambiente de negocios turbulento.
- II. Agilidad es la habilidad para balancear flexibilidad y estabilidad.

Basándose en las características anteriores, dan como resultado que la agilidad de los métodos de desarrollo de software se puede sintetizar en la capacidad para responder a los cambios y para simplificar los procesos (Orjuela Duarte & Rojas C., 2008).

Con base, en lo anterior y a la experiencia en el uso de una metodología ágil de desarrollo de software móvil, se ha formulado una definición propia, siendo la siguiente:

“Una METODOLOGÍA ÁGIL DE DESARROLLO DE SOFTWARE es el eficiente enfoque de adaptación y en algunos casos innovación. De un conjunto disciplinado, ágil y sistemático uso de técnicas, procedimientos, prácticas, estados, usos, costumbres, documentos e información, que permiten un enfoque de observación rigurosa y análisis total de un problema asunto y/o objeto, empleados para la correcta gestión del desarrollo de un producto de software (sistema / app) con alta calidad, haciendo las liberaciones justas del producto; conjuntamente con el firme y único objetivo de lograr la satisfacción del cliente, maximizando el retorno rápido de la inversión y a la vez tomando una acertada decisión”.

2.1.1.2. Principios de las metodologías Ágiles.

En febrero y marzo del 2001, tras una reunión celebrada en Utah-EEUU., nace el término *Ágil*, aplicado al desarrollo de software. Diecisiete críticos de los modelos de mejora de desarrollo de software basados en procesos, convocados por Kent Beck, quien había publicado un par de años antes *Extreme Programming Explained*, libro en el que exponía una nueva metodología denominada *Extreme Programming*, se reunieron en Salt Lake City, para tratar sobre técnicas y procesos para desarrollar software (Fowler & Highsmith, 2001).

El principal objetivo, de la mencionada reunión, fue diseñar los valores y principios que deberían permitir a los equipos desarrolladores, crear productos de software rápidamente y respondiendo adecuadamente a los cambios que pudieran surgir a lo largo del proceso de desarrollo de un proyecto. En dicha reunión se creó el término “Métodos Ágiles”, para referirse al uso de los métodos que estaban surgiendo como alternativa a las metodologías tradicionales (CMMI, SPICE y demás), mismas a las que se consideraban excesivamente pesadas y rígidas por su carácter normativo y fuerte dependencia a las planificaciones detalladas previas al desarrollo.

Como resultado, surgió *The Agile Alliance*, como una organización y/o comunidad, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software. Ayudando al otorgar un voto de confianza, y de esta manera fomentar que las

organizaciones adopten dichos conceptos, mostrando un gran interés de ponerlos en práctica. Los presentes en dicha reunión colaboraron, con base en su experiencia y conocimiento a definir los principios sobre los que se basarían, estos métodos alternativos. Resultando en cuatro postulados, los que ha quedado, el denominado documento llamado: el Manifiesto Ágil, que actualmente resume la filosofía de desarrollo ágil.

Los valores inicialmente honrados por los agilistas Beck, Fowler y Highsmith, se presentan en lo general a continuación:

- Individuos e interacciones sobre procesos y herramientas.
- Software de trabajo sobre documentación completa.
- Colaboración del cliente sobre la negociación del contrato.
- Responder al cambio sobre el seguimiento de un plan.

No obstante, el manifiesto ágil está fundamentado en las siguientes cualidades, explicadas a detalle y que se expresan a continuación:

- I. Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. Las personas son el principal factor de éxito de un proyecto de software. Es más importante construir un buen equipo de trabajo, que construir el entorno. Muchas veces se comente el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- II. Desarrollar software que funciona más que conseguir una buena documentación. La regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental.
- III. La colaboración con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- IV. Responder a los cambios más que seguir estrictamente un plan. La habilidad rápida de responder a los cambios que puedan surgir a lo largo del proyecto

(ya sea en los requisitos, en la tecnología, en el equipo, por mencionar algunos) es otro factor que determina el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

Las aptitudes anteriores inspiran, los doce principios del manifiesto ágil. Son estas características, las que diferencian un proceso ágil de uno tradicional. Se considera que los dos primeros principios son el resumen en gran parte, del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto a metas a lograr y organización del mismo.

Los principios centrales del manifiesto ágil son:

1. *La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte valor.*
2. *Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.*
3. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
4. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
5. Construir el proyecto entorno a individuos motivados. Darles el entorno y que necesitan y confiar en ellos para conseguir finalizar el trabajo.
6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
7. El software que funciona es la medida principal de progreso.
8. Los procesos ágiles promueven un desarrollo sostenible.
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requisitos, diseños surgen de los equipos organizados por sí mismos.
12. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivos, y según esto ajusta su comportamiento (Fowler & Highsmith, Op.Cit.).

2.1.1.3. Diferencia entre metodologías ágiles vs. Metodologías tradicionales.

Un método de desarrollo de software funciona mejor cuando se aplica en la o a las situaciones con características muy específicas (llamadas “*HomeGround*” -origen y/o bases), del proceso de desarrollo. En la Tabla 1., se puede observar la comparación general entre las bases teóricas de los métodos ágiles y las de los procesos de desarrollo tradicionales o también llamados por planes o “*plan-driven*” (planificado).

Tabla 1. Diferencias entre metodologías ágiles vs. Metodologías tradicionales.

ÁREA	METODOLOGÍA ÁGIL	MÉTODOS TRADICIONALES
Desarrolladores	Colaborativos, unidos, ágiles y entendidos	Orientados al plan con una mezcla de habilidades
Clientes	Son representativos y se les entrega poder	Mezcla de niveles de aptitud
Confianza	Conocimiento tácito interpersonal	Conocimiento explícito documentado
Requerimientos	En gran parte emergentes y con rápidos cambios	Conocibles tempranamente y bastante estables
Arquitectura	Diseñada para los requerimientos actuales	Diseñada para los requerimientos actuales y los del futuro próximo
Refactorización	Económica	Costosa
Tamaño	Productos y equipos pequeños	Productos y equipos más grandes
Premium	Valor rápido	Alta seguridad

Fuente: Blanco, Op. Cit., 2009.

El desarrollo ágil de software, pretende ser una práctica que se lleve con facilidad, con mucha ligereza de procesos y/o documentación; evitando lo complejo y el lento andar de las metodologías tradicionales, poniendo un justo énfasis cada vez más necesario en las personas (desarrolladores) y los resultados (artefactos), resultando en una reducción significativa del tiempo de desarrollo en sí mismo producto. Las metodologías ágiles de desarrollo promueven interacciones constantes (internas y externas), durante el proceso de desarrollo del ciclo de vida del proyecto, creando software, en muy cortos lapsos de tiempo (iteraciones), mismas que minimizan los riesgos, las cuales pueden tener una duración máxima, de una a cuatro semanas, con base a las características propias del proyecto.

Las iteraciones no tienen la intención de agregar demasiada funcionalidad y con ello justificar un temprano lanzamiento y/o liberación del producto de software al mercado, pero

si tienen el firme objetivo de conseguir una versión útil, de la implementación funcional de uno y/o varios requerimientos iniciales, sin errores. Lo que conlleva a que el equipo desarrollador, tenga la necesidad de volver a evaluar, las prioridades del proyecto con mayor frecuencia, encontrando un cambio, más óptimo, lo que logrará un software de alta calidad, debido a la innovación en el modelo de negocio.

Como se puede observar en la Tabla 2., las metodologías ágiles poseen ciertas propiedades que las hacen totalmente aplicables, al ámbito del desarrollo de software móvil “Apps”. Dichas características son las ideales y una posible motivación en cada caso, como se muestra a continuación

Tabla 2. Comparativa entre las características básicas o bases (*home ground*) ágiles y los rasgos observados en el desarrollo de software móvil.

CARACTERÍSTICAS ÁGILES	MOTIVACIÓN LÓGICA	DESARROLLO MÓVIL
Alta Volatilidad del Entorno	Debido a la alta frecuencia en el cambio, que sufren los requerimientos tendremos menos necesidad de diseño y planificación inicial y mayor necesidad de desarrollos incrementales e interactivos.	<ul style="list-style-type: none"> Alta incertidumbre Entornos dinámicos Cientos de nuevos terminales cada año.
Equipos de Desarrollo Pequeños	Capacidad de reacción más rápida, trabajo basado en la compartición de la información, menos documentación.	<ul style="list-style-type: none"> La mayor parte de los proyectos de desarrollo de software para plataformas móviles se lleva a cabo en microempresas y PYMES.
Cliente Identificable	Desaparecen los malentendidos.	<ul style="list-style-type: none"> Potencialmente, hay un número limitado de usuario finales, pero los clientes son fáciles de identificar.
Entornos de Desarrollo Orientados a Objetos	Mayoría de las herramientas de desarrollo ágil existen bajo plataformas orientadas a objetos.	<ul style="list-style-type: none"> Por ejemplo, Java y C++, se usan, algunos problemas en herramientas como refactorizaciones o primeros test.
Software Crítico No Asegurado	Los fallos no causan gran impacto, como la pérdida de vidas. Se puede buscar mayor agilidad en el desarrollo.	<ul style="list-style-type: none"> La mayoría del software es para entretenimiento. Los terminales no son fiables.
Software a Nivel de Aplicación	Sistemas embebidos grandes requieren comunicación exhaustiva y mecanismos de verificación.	<ul style="list-style-type: none"> Mientras los sistemas móviles son complejos y altamente dependientes Las aplicaciones son muy autónomas.
Sistemas Pequeños	Menos necesidad de diseño inicial.	<ul style="list-style-type: none"> Las aplicaciones, aunque variables en tamaño no suelen superar las 10,000 líneas de código.
Ciclos de Desarrollo Cortos	Propósito de realimentación rápida.	<ul style="list-style-type: none"> Periodos de desarrollo de 1 a 6 meses.

Fuente: Blanco, Op.Cit., 2009.

2.1.1.4. Escenario general de trabajo de una metodología ágil.

Conjuntamente, con todo lo anteriormente descrito, no se puede dejar de lado, ni olvidar, los escenarios y/o entornos precisos, de trabajo que siempre utilizan las metodologías ágiles de desarrollo de software, llamadas dentro del argot informático *ceremonias*, con la finalidad de incrementar la retroalimentación y facilitar la coordinación de los equipos de desarrollo; logrando mayores y mejores niveles de integración (relación) entre de los mismos:

1. PLANIFICACIÓN CONFORME A VALORES *“PLANNING POKER”*

El nombre es tomado, porque se utilizan cartas numeradas (no precisamente de póker), mejor y más correctamente expresada, es el cálculo de la estimación del esfuerzo, es una técnica muy sencilla, divertida y eficaz (donde lo normal es numerar las cartas con una serie de Fibonacci, más la carta interrogante que equivale a un “ni idea” y la carta infinito que equivale a “esto es demasiado grande”), para el desarrollo de los entregables requeridos por el cliente; en el que todas las personas comprometidas en el desarrollo de un producto de software, opinan para estimar el esfuerzo de desarrollar cada una de las historias de usuario de una sesión “sprint” (Romeu, 2014).

2. REUNIONES DIARIAS *“DAILY MEETINGS”*

Son reuniones diarias rápidas para la coordinación del equipo de desarrollo, lo que permite una aceleración sinérgica, integridad del equipo, aumenta la productividad, en el entendimiento del trabajo a realizar y genera una mejora continua orientada a la calidad del producto de software, una efectiva reunión diaria debe cumplir estrictamente, los siguiente cinco puntos: 1) Debe ayudar a empezar bien el día, con fuerza y energía, 2) Debe ayudar a comunicar lo que está pasando, 3) Debe ayudar a resolver y compartir los problemas, 4) Debe ayudar a reforzar el enfoque hacia los objetivos correctos y 5) Debe ayudar a reforzar el sentido de equipo (Garzás, 2015).

3. DEMOSTRACIÓN DE REQUISITOS COMPLETADOS POR SESION

“SPRINT DEMO”

Se considera una sesión y/o reunión informal donde el equipo desarrollador presenta al cliente, los requisitos completados en la iteración anterior, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. Como una retroalimentación entre ambos, en función de mostrar resultados obtenidos y de los cambios que haya habido en el contexto del proyecto. El cliente realiza las adaptaciones necesarias de manera objetiva, desde la primera iteración, replanificando constantemente el proyecto. La duración máxima de demostración es de cuatro horas (proyectos ágiles.org, s.f.).

4. REUNIÓN DE RETROSPECTIVA “*SPRINT RETROSPECTIVE*”

La reunión de retrospectiva se produce después de la demostración de requisitos completados y antes de la planificación de la próxima sesión de tareas (*sprint*). Generalmente es una reunión de tres horas máximo para discutir los sprints de un mes. En la reunión de retrospectiva, el equipo desarrollador debe discutir las siguientes tres premisas: 1) Lo que salió bien en el *Sprint* anterior, 2) Lo que podría mejorarse y 3) Qué se comprometen a mejorar en el próximo *Sprint* (Scrum.org, 2017).

2.1.1.5. Factores que afectan la agilidad de un proceso de desarrollo de software.

Generalmente, durante el proceso de desarrollo de un proyecto software, suelen presentarse cinco factores principales que afectan considerablemente la agilidad del mismo:

1. *Ausencia de una cultura de trabajo en equipo*: la problemática se presenta cuando dentro de los miembros del equipo desarrollador y/o trabajo; no se entiende, ni se cree en el pensamiento de trabajo en equipo, además de que no se realizan en forma corporativa la planificación, la toma de decisiones y las actividades pertinentes, no se reconoce la creencia básica “ninguno de nosotros es tan bueno sólo, como todos nosotros juntos”.
2. *El tamaño de equipo de desarrollo*: la problemática se presenta cuando el número de miembros del equipo desarrollador y/o trabajo, es bastante grande, ya que cada uno con sus responsabilidades y tiempos, complica que se puedan llevar a cabo reuniones completas y cortas.
3. *La complejidad del producto de software*: la problemática se presenta cuando, la complejidad, de “un atributo” (conformado por las características específicas que tenga que cumplir el producto de software y el costo tiempo / esfuerzo), afecta directamente, y fomenta que el proyecto pueda estar más propenso a errores.
4. *Experiencia y/o preparación técnica de los desarrolladores*: la problemática se presenta cuando dentro del equipo desarrollador de planta, no existe el recurso necesario para crear el producto de software solicitado, y esto genera que se adhieran nuevos miembros al equipo desarrollador, lo que puede denotar un instinto de amenaza, en muchos factores, dentro del mismo equipo.

5. *La correcta comprensión colectiva de los requerimientos:* la problemática se presenta cuando, no existe una comprensión correcta y específica de todos los requerimientos, lo que otorga al proyecto cierto grado de vacío y genera demasiados cambios, así mismo, muy constantes.

2.1.1.6. Ventajas del uso de las metodologías ágiles.

Las metodologías ágiles aportan beneficios y ventajas, siendo una de estas, la gestión del proyecto, mediante prácticas ligeras, otorgando cuatro garantías principales:

- Valor.
- Productividad (Reducción de tiempo, agilidad y fiabilidad).
- Garantizar la calidad.
- La satisfacción total del cliente.

Además, de que se realiza un seguimiento diario de los avances del proyecto, logrando que los integrantes del equipo de desarrollo estén unidos, comunicados y que el cliente vea un indiscutible avance y continuidad, tangible, lo que le genera seguridad.

El uso de metodologías ágiles de desarrollo de software asigna las siguientes cualidades y virtudes:

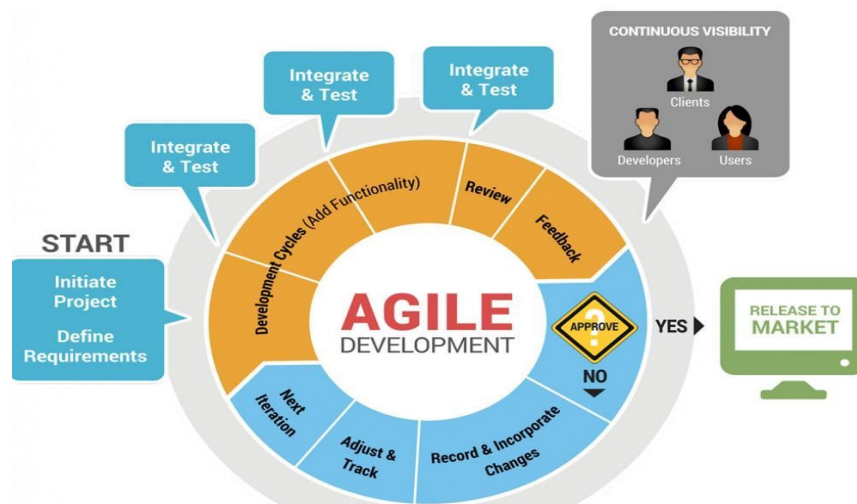
1. Adaptación (respuesta) rápida a los cambios, en los requisitos iniciales del proyecto, implementando soluciones sobre la marcha.
2. Eliminación de tareas innecesarias, lo que minimiza los costos de desarrollo, ya que se priorizan las características y funcionalidades del sistema, unificando criterios de acción.
3. Mayor intervención activa del cliente durante el proceso, y en cada una de las etapas, aportando ideas, opiniones y tomando acertadas decisiones del mismo.
4. Mayor velocidad y eficiencia en los intervalos de entrega, las entregas parciales o en bloques mejoran la optimización de recursos y optimizan las labores de seguimiento y control.

5. Identificación y resolución de errores con mayor facilidad, debido a que se van haciendo constantes pruebas a medida que se avanza, en el desarrollo del producto de software.
6. Comunicación constante, la totalidad del equipo desarrollador, conoce el estado del mismo, en todas las etapas del proceso de desarrollo del proyecto.
7. Simplificación de la sobrecarga de procesos, el equipo desarrollador trabaja sobre normas y regulaciones, lo que mejora considerablemente la calidad final del producto de software.
8. Generación de alta calidad, las constantes interacciones entre el equipo desarrollador y el cliente; proporcionando la suficiente satisfacción de expectativas del cliente y seguridad.

2.1.1.7. Metodologías ágiles para el desarrollo de software móvil.

En el vasto universo del desarrollo de software móvil, actualmente existen diversas metodologías ágiles para su desarrollo, cada una, con sus puntos fuertes y sus puntos débiles, respectivamente. Sin embargo, cuando se plantea la necesidad de elegir una, se torna un tanto complicado, por lo que se recomienda escogerla, dependiendo el entorno (número de desarrolladores) y en función de las necesidades propias del proyecto de desarrollo (requerimientos), como lo ilustra en lo general la Figura 1.

Figura 1. Metodologías ágiles.



Fuente: Carlos Rodríguez, 2015.

Para este momento, en la presente investigación científica tecnológica, se tiene conocimiento de las múltiples metodologías ágiles existentes para el desarrollo de software como los son: CC, DSDM, XP, FDD, RUP, SCRUM, Agile Modeling, Agile Model-Driven Development, 6 Sigma, J2ME, por mencionar algunas.

A pesar de existir muchas más estas son las que disponen de mayor documentación. No obstante, todas las antes mencionadas, no se centran en los requerimientos y particularidades específicas que la creación de un producto de software móvil necesita.

A continuación, se especificarán en lo general algunas metodologías ágiles para Apps, con mayor documentación e información, así como sus particularidades:

→ **SXP-J2ME**

Es un método híbrido de las bien conocidas XP y SCRUM, surgió en abril del 2012, con el objetivo de diseñar una metodología de desarrollo de software que se adecuará a las características de los proyectos que conforman la línea de productos de aplicaciones J2ME para la cultura y el patrimonio, pertenecientes a la facultad regional de la Universidad de las Ciencias Informáticas de Granma. La línea antes mencionada la componen proyectos que utilizan la plataforma Java 2, (*Micro Edition 8J2ME*, por sus siglas en inglés), para el desarrollo de Apps y están enfocados en la promoción y difusión de la cultura y el patrimonio de la provincia Granma (Domenech & Abad, 2013).

→ **DELFDROID**

En la metodología delfdroid se proponen un total de ocho roles: equipo del proyecto (*Team*), jefe del proyecto (*Project Manager*), jefe de producto (*Product Management*), arquitecto (*Architect*), visionario (*Visionary*), add (*Analyst Designer Developer*), encargado de pruebas (*Tester*) y consultor (*Consultant*). Además, cuenta con tres fases, las mismas indican el énfasis propuesto por Barry Boehm en relación a la división en dos perspectivas de desarrollo. Se hacen prospectos de proyecto, se investigan proyectos similares y su factibilidad económica. Si durante el proceso o las etapas se llega a una versión estable se procede a registrarlo (Domenech & Abad, 2013).

→ **DMAMC**

El método aplicado, que se denomina definir, medir, analizar, mejorar y controlar (*DMAMC*, por sus siglas en el idioma inglés), utiliza herramientas estadísticas, además de dispositivos que observan las variables de los procesos y sus relaciones, que ayudan a gestionar sus características. El método consiste en la aplicación, proyecto a proyecto de un proceso estructurado en cinco fases, por un lado, mejorar las características del producto, permitiendo conseguir mayores ingresos, el ahorro de costos que se deriva en la disminución de fallos o errores y de los menores tiempos de ciclo en los procesos (GestioPolis.com, 2001).

→ **MADAMDM**

Para elaborar esta metodología se utilizaron cuatro distintas metodologías las cuales son: *XP*, Metodologías Orientadas a Objetos para la producción de Software Multimedia e Hipermedia (*MOOMH*, por sus siglas en el idioma inglés), basado en ingeniería Web (*UWE*, por sus siglas en el idioma inglés) y método de Diseño Hipermedia Objeto Orientado (*OOHDM*, por sus siglas en el idioma inglés), agarrando características relevantes que ayudan a concretar los objetivos y a facilitar el trabajo de desarrollo y el diseño de la app, en varios sistemas operativos (Fuzi, 2013).

→ **MOBILE- D**

La metodología ágil apunta a la necesidad de disponer de un ciclo de desarrollo muy rápido para equipos muy pequeños (está pensado para grupos de no más de 10 desarrolladores máximo). Los proyectos se deberían finalizar con el lanzamiento de productos completamente funcionales en menos de diez semanas. Todas las fases, con excepción de la primera fase exploratoria, contienen tres días de desarrollo distintos: planificación, trabajo y liberación; añadiendo días para acciones en casos particulares (Escobar Bonilla & Campaña Guzmán, 2014).

→ **MODELO WATERFALL (EN CASCADA)**

La metodología clásica de cascada, solo es aplicable cuando están totalmente cerrados los requisitos y no van a cambiar. No hay retroalimentación entre las fases, en las que se divide el proyecto. Por lo que cada una de las fases se va cerrando de forma secuencial. Todo el proceso está fijado por fechas límites y presupuestos. Este

modelo solo es aconsejable para proyectos móviles muy controlados y previsibles, no existe incertidumbre por lo que se quiere hacer, ni influyen los cambios en la industria (Rodríguez , 2011).

→ **MOBILE DEVELOPMENT PROCESS SPIRAL**

Se basa en el desarrollo de fases sucesivas, por evolución de sistemas más simples a sistemas más complejos. Además, utiliza el modelo de desarrollo en espiral como base, e incorpora procesos de evaluación de usabilidad, priorizando la participación del usuario en todos los procesos del ciclo de vida de diseño. También está orientado a proyectos grandes y costosos, ya que está diseñado a ser un modelo de reducción de riesgos (Schenone Marcelo, 2004).

→ **DESARROLLO RÁPIDO DE APLICACIONES**

El método de desarrollo rápido de aplicaciones es un modelo de desarrollo interactivo cuyo objeto es conseguir prototipos lo antes posible para mejorarlos después poco a poco. Se suelen priorizar la implementación sobre la planificación, y se utilizan muchos patrones de diseño conocidos para poder adaptarse de la mejor manera a los cambios en los requerimientos. El desarrollo rápido de aplicaciones es un método muy útil para el desarrollo de proyectos regularmente urgentes con tiempos de entrega muy cortos (Luna Perez , 2016).

→ **DESARROLLO ÁGIL**

El desarrollo ágil es un método de desarrollo basado en iteraciones, donde en cada iteración se realizan todas las fases del ciclo de desarrollo. El desarrollo ágil se basa en los principios del manifiesto ágil y sus valores éticos, que tratan de dar más valor a algunos conceptos, pero sin dejar de lado los demás. Con estos valores e intenta conseguir, entre otras cosas, entregar algo lo más pronto posible y evitar problemas originados por cambios de requisitos. Esto es muy apropiado para proyectos cambiantes, ya sean grandes o pequeños, ya que mediante estos valores pueden mitigar los riesgos (Luna Perez, Op. Cit., 2016).

→ **METODOLOGÍA DE DISEÑO HÍBRIDO**

Se apoya en una combinación del desarrollo adaptativo de software (*ASD*, por sus siglas en el idioma inglés) y el diseño de nuevos productos (*NPD*, por sus siglas en

el idioma inglés). Esto supone una decisión crítica para decantarse más al lado del desarrollo de productos que del lado de la gestión de proyectos, lo cual quiere decir que una de las características más sensibles, desde el punto de vista metodológico, para la consolidación de una metodología propia de un entorno móvil, es la presión de los plazos para llegar al mercado, un mercado volátil y altamente dinámico (Blanco, Camarero, Fumero, Warterski, & Rodríguez, 2009).

→ **KANBAN**

La metodología Kanban tiene su significado de las palabras japonesas que significan “tarjetas visuales”, donde Kan es “visual” y Ban corresponde a “tarjeta”. Las principales ventajas de esta metodología, es muy fácil de utilizar, actualizar y asumir por parte del equipo desarrollador, se basa, en una serie de principios que la diferencian del resto de metodologías conocidas como ágiles: CALIDAD GARANTIZADA, todo lo que se hace debe salir bien a la primera, no hay margen de error. De aquí que en Kanban no se premie la rapidez, sino la calidad final de las tareas realizadas; REDUCCIÓN DE DESPERDICIO, se basa en hacer solamente lo justo y necesario, pero hacerlo bien; MEJORA CONTINUA, Kanban no es simplemente un método de gestión, sino también un sistema de mejora en el desarrollo de proyectos, según los objetivos a alcanzar (Gilibets, 2013).

→ **DEVOPS**

Es una serie de prácticas, como lo son la participación efectiva de los administradores de sistemas en el proceso de desarrollo de aplicaciones, utilizando las mismas técnicas ágiles que usan los desarrolladores. *DevOps* significa un montón de cosas diferentes para diferentes personas porque el “desarrollo ágil” en sí mismo cubre un terreno bastante amplio. Algunas personas dirán que *DevOps* es la colaboración de los departamentos de desarrollo y operaciones, o que es tratar el código como infraestructura o usar automatización, o una variedad de términos similares que se relacionan de una forma u otra (Patricio, 2013).

→ **SCRUMBAN**

Es una metodología ágil que combina las mejores características de los métodos SCRUM y KANBAN, permitiendo a los equipos moverse hacia el desarrollo ágil y ayuda a mejorar constantemente sus procesos. *Scrumban* se está volviendo muy

popular en industrias en las que el desarrollo del proyecto y el mantenimiento van juntos. Ayuda al equipo de desarrollo a eliminar el elevado estrés, mejora la eficiencia y la satisfacción general del cliente (Labs, 2009-2017).

2.2 INGENIERÍA DE SOFTWARE MÓVIL

2.2.1. Concepto de Ingeniería de software.

La ingeniería de software es una disciplina de la ingeniería que comprende, todos los aspectos de la producción de software desde las etapas iniciales desde la especificación de sistema, hasta el mantenimiento de éste, después de su uso.

En esta definición, existen dos frases claves:

1. *Disciplina de la ingeniería*: Los ingenieros hacen que las cosas funcionen, aplican teorías, métodos y herramientas, donde sean convenientes, pero las utilizan de forma muy selectiva, siempre tratando de descubrir soluciones a los problemas, aun cuando no existan teorías y métodos aplicables para resolverlos. Los ingenieros también saben que deben trabajar con restricciones financieras y organizacionales, por lo que buscan soluciones tomando en cuenta estas restricciones.
2. *Todos los aspectos de la producción de software*: La ingeniería de software no sólo comprende los procesos técnicos del desarrollo de software, sino también con actividades tales como la gestión de proyectos de software y el desarrollo de herramientas, métodos, teorías y de apoyo a la producción de software.

En general, los ingenieros de software adoptan un enfoque sistemático y organizado en su trabajo, ya que es la forma más efectiva de producir software de alta calidad. Sin embargo, aunque la ingeniería de software consiste en seleccionar el método más apropiado para un conjunto de circunstancias, un enfoque más informal y creativo de desarrollo podría ser efectivo sólo, en algunas únicas circunstancias (Sommerville, 2005).

La ingeniería de software, trata con áreas muy diversas de la informática, de las ciencias de la computación e ingeniería de requerimientos tales como: la construcción de compiladores,

los sistemas operativos o desarrollos Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información, aplicables a una infinidad de áreas tales como: los negocios, la investigación científica, la medicina, la producción, logística, la banca, el control de tráfico, la meteorología, el derecho, Internet, Intranet, solo por mencionar algunas.

Desafortunadamente no existe una definición precisa sobre la ingeniería de software completa en los diccionarios. Sin embargo, se pueden citar las enunciadas por algunos de los más prestigiosos autores del gremio:

- La Ingeniería de Software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software (Zelkowitz, 1978).
- La Ingeniería de Software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software (Bohem, 1976).
- La Ingeniería de Software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales (Bauer, 1972).
- La Ingeniería de Software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software (IEEE, 1993).

Algunos autores, consideran que el desarrollo de software, es el término más apropiado, que ingeniería de software, para referirse al proceso de creación de software. Personas como, Pete McBreen autor de "*Software Craftsmanship*", cree que el término ingeniería de software, implica niveles de rigor y prueba de procesos que no son apropiados, para todo tipo de desarrollos de software.

La creación de software es un proceso intrínsecamente creativo y la ingeniería del software, es la que trata de sistematizar este proceso, con el fin de acotar el riesgo, el fracaso en la consecución del objetivo creativo, por medio de diversas técnicas que se han demostrado adecuadas, con base en la experiencia previa.

La ingeniería de software se puede considerar como la ingeniería aplicada al software, esto es, por medios sistematizados y con herramientas preestablecidas, la implementación de ellos, de la forma más eficiente para la obtención de resultados óptimos; siendo estos los objetivos, los que siempre busca la ingeniería. No es sólo la resolución de problemas, sino más bien tener en cuenta las diferentes soluciones y elegir la más apropiada (Rodlt, 2012).

Con base en lo anterior, se ha formulado una propia definición:

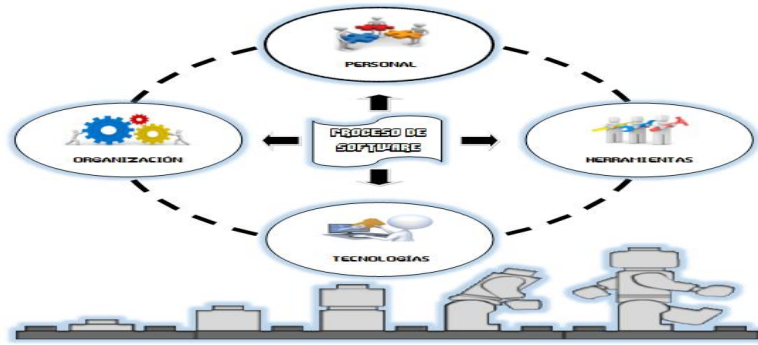
“La INGENIERÍA DE SOFTWARE y/o DESARROLLO DE SOFTWARE es la especialidad, que divide el trabajo en distintas actividades y/o tareas, con la firme intención de lograr siempre la mejor gestión de todos los recursos disponibles y obtener el resultado más eficiente posible para la creación de un producto de software, de su mejoramiento, así como de su evolución, en el caso de un producto ya existente, mediante el proceso de aplicación de programas y conocimientos informáticos, en una serie de actividades y/o principios disciplinarios, que permiten el uso optimizado de los modelos, métodos, metodologías, técnicas y/o prácticas con el objetivo de construir, manipular (funcionamiento) y mantener un producto de software de alta calidad”.

2.2.2. El proceso de desarrollo del software.

Un proceso de desarrollo software define: ¿Quién?, está haciendo; ¿Qué?; ¿Cuándo? y ¿Cómo?, y de esta manera alcanzar un determinado objetivo de creación; con la ingeniería de software, con el propósito efectivo de proporcionar las normas para el progreso eficiente de un producto de software, con alta calidad que el estado vigente de la tecnología permita.

De esta forma, un proceso de desarrollo de software debe ser capaz de evolucionar con los años y con la experiencia. Durante ésta y en un momento determinado de tiempo, debe limitar su alcance, a las realidades que permitan las tecnologías, herramientas y participantes de una organización, en paralelo. Un generador del proceso de desarrollo de software, tiene la responsabilidad de acotar un conjunto de conocimientos; necesarios para ser ejecutados con las capacidades y/o habilidades actualmente existentes en los miembros del equipo de desarrollo en una organización, como lo ilustra la Figura 2. (Álvarez, y otros, 2014).

Figura 2. El proceso de software y su evolución.



Fuente: Elaboración Propia y LEGO Evolución, 2019- 2017.

2.2.3. Concepto de Ingeniería de software móvil.

Siempre que se tenga la intención o motivación, de crear una App de alta calidad, el equipo desarrollador, deberá utilizar ingeniería de software móvil, con la finalidad de lograr la satisfacción plena de los requerimientos demandados por los usuarios y a la vez permitir una sencilla instalación, actualización y mantenimiento, por un largo periodo de tiempo.

MÓVIL, significa que puede moverse o se mueve por sí mismo, ya que no tiene estabilidad o permanencia, en un solo espacio determinado (Real academia española. Op. Cit.).

La ingeniería de software móvil implica que el proceso de desarrollo (modelo, método, metodología y/o técnica), que se empleó para la creación de una un producto de software móvil (App), denominado formalmente como ciclo de vida estándar del mismo, se encuentra plenamente conformada por siete etapas y/o estados generales, establecidos en la Tabla 3., de la siguiente manera:

Tabla 3. Etapas del Proceso de Desarrollo de Software Móvil.

ETAPAS DEL PROCESO DE DESARROLLO DE SOFTWARE MÓVIL	
Etapa	Descripción
Concepción	En esta etapa, se definen claramente "el o los" problemas que se desean resolver (concepción de la idea), fijando claramente los objetivos del sistema "Aplicación Móvil", que se desea crear. Identificando los componentes principales que integran al producto (comprendiendo claramente el y/o los modelos de negocio); desarrollando el modelo, método y/o metodología, que se seguirá, en la totalidad del proceso y que se adapte plenamente a las necesidades del equipo desarrollador del proyecto de software.
Confección	En esta etapa se establecen las características (que debe y que no debe, contener el sistema y/o app) y cómo será arquitectura del mismo y ¿Por qué?, la principal tarea es desarrollar producto (sistema / app) que cumpla todas las especificaciones (requerimientos) ya fueran iniciales, temporales y/o finales, priorizándolos racionalmente.
Construcción	En esta etapa consiste en utilizar los modelos creados, las herramientas de desarrollo físicos y/o lógicos, simuladores físicos y/o lógicos, lenguajes de programación necesarios, básicamente todo lo que implica el desarrollo del software; para crear los componentes y lograr las funcionalidades que requiera el sistema.
Lanzamiento	En esta etapa es el momento en el cual se transfiere y pone a disposición del usuario final y/o cliente, el producto de software (sistema / app) final.
Mantenimiento	En esta etapa se proponen habitualmente actualizaciones, mejoras y/o cambios, incluso en esta etapa se pueden desarrollar nuevos elementos, funcionalidades y/o componentes, con la misión de subsanar y corregir los problemas (errores) aparecidos en el producto, así como lograr un mejor rendimiento del mismo.
Calidad	En esta etapa se propone el aseguramiento de la calidad del software del sistema "Aplicación Móvil", la cual tiene que tener en cuenta, los siguientes factores generadores de complejidad y que radican en la existencia de diferentes sistemas operativos y la multitud interminable de modelos de dispositivos móviles existentes para su ejecución. De esta forma, una app debidamente construida y de calidad será aquella que funcione correctamente, en todos los dispositivos móviles, considerando las características propias de los mismos y bajo todos los sistemas operativos existentes en el mercado. Así mismo, la app debe ser capaz de guardar, tanto de forma efectiva como selectiva, la información, dejando en el servidor y/o memoria interna del dispositivo aquellos datos que precisen de recursos adicionales, observando su comportamiento y errores que pueda presentar cuando el ambiente cambie, por ejemplo, batería baja, entrada de llamadas o mensajes, mientras se ejecuta la operación de la misma.
Capacitación	En esta etapa final del proceso consiste en realizar todas las tareas necesarias para asegurar que los clientes, empleados y/o usuarios finales, estén conscientes del funcionamiento del sistema y reflexionar que el producto "ya no será vendido, ni soportado", de aquí en adelante.

Fuente: Elaboración Propia, 2017.

Por lo cual se vuelve de trascendental importancia, seguir al pie de la letra y de manera más clara, todas las etapas generales de un ciclo de vida estándar del software móvil (desde la concepción de la idea, análisis de requerimientos, diseño, desarrollo, lanzamiento, pruebas, cambios, publicación, capacitación de uso y mantenimiento), pero teniendo muy presente, las vastas diferencias que existen actualmente entre el desarrollo de una App, para ejecutarse en una computadora de escritorio⁶ y el de una App, para ejecutarse en un dispositivo móvil.

⁶ Una aplicación de informática de escritorio (Plugin, procedente del idioma inglés, cuyo significado es "enchufable"), es aquella que añade funcionalidades específicas a un programa principal. Su presencia es mucho más habitual en los navegadores Web, en los reproductores de música y en los sistemas de gestión de contenidos que se utilizan en una computadora de escritorio, desde los años 70's, bajo el nombre de: complementos, extensiones y addons, en inglés, add-on "agregado", (Saberia.com, 2012).

2.3. DEFINICIÓN DE CALIDAD.

2.3.1. Concepto de calidad.

La calidad engloba todo el proceso de construcción de un producto o prestación de servicio y está determinada por factores directos e indirectos. Aunque la calidad es un concepto amplio, complejo, multifacético, subjetivo y muchas veces ambiguo, debido a que este puede describirse desde diferentes perspectivas. Es por esto que no es posible estandarizar su definición y satisfacer todos los criterios de una manera simple y efectiva.

Sin embargo, se puede trabajar sobre las mejores definiciones reconocidas, siendo estas las siguientes:

- *Según la Real Academia de la Lengua Española (RAE), en su versión electrónica (DLE):* La calidad es la propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor, como de buena calidad, superioridad o excelencia, adecuándose a las características especificadas (condiciones o requisitos) que se ponen en un contrato.
- *Según el modelo de la norma ISO 9000:* La calidad es el grado en el que un conjunto de características inherentes cumple con los requisitos, entendiéndose por requisito, a la necesidad o expectativa establecida, generalmente implícita u obligatoria.
- *Según el modelo de la norma ISO 9001:* propone, con base en la *ISO 9000:* Un enfoque de calidad basada en un sistema conformado de múltiples elementos (estos elementos son los diversos procesos que se llevan a cabo dentro de una organización [entiéndase empresa, organismo público, asociación, sea cual sea su actividad, su tamaño o ubicación geográfica), interrelacionados entre sí (o que interactúan)], y cuya gestión de manera definida, estructurada y documentada, debe permitir lograr un nivel de calidad que alcance la satisfacción del cliente, objetivo final de este modelo

2.3.2. Problemáticas para definir la calidad.

Son innumerables las problemáticas que se presentan para definir a la calidad en cualquier área de trabajo. La ingeniería de software no escapa de esta cruda realidad, pero sin lugar a dudas se enfrenta con muchos obstáculos, para poder determinar la calidad en

comparación con otras ciencias y delimitar un dominio de la misma. Se pueden enumerar solo algunos de estos a continuación:

- *No existe una definición estándar y universal de qué es calidad:* En realidad algunos organismos e instituciones como la Organización internacional para la estandarización (*ISO*, por sus siglas en el idioma inglés); el Instituto de ingenieros eléctricos y electrónicos (*IEEE*, por sus siglas en el idioma inglés); el Instituto de ingeniería de software (*SEI*, por sus siglas en el idioma inglés); el Instituto Europeo del software (*ESI*, por sus siglas en el idioma inglés); entre muchos otros organismos, brindan definiciones aceptables, pero no son homogéneas, debido a que la calidad no es como tal una idea o una frase únicamente; es un concepto que se tiene que observar desde varias dimensiones, por lo que existen diferentes niveles de comprensión, por lo que es de suponer que varíe para cada persona en lo particular, dando, como resultado que cada profesional utilice su propia versión y/o definición de calidad (Álvarez, y otros, 2014).
- *La calidad debe satisfacer a una amplia gama de entes relacionados, pero no mutuamente excluyentes:* clientes, procesos, organismos, productos (documentos, aplicaciones, mediciones, por mencionar algunas), la magnitud de calidad incluye la utilidad del producto, la perspectiva del producto, y los atributos propios de calidad del producto, por lo que son muchos los factores a considerar para determinarla correctamente (Álvarez, y otros, Op. Cit.).
- *Crear una cultura de calidad:* Esto implica un compromiso constante, tedioso, costoso, y a largo plazo por parte de la organización y las personas que lo componen, de ceñirse a la creencia de calidad (Álvarez, y otros, Op. Cit.).

2.3.3. Definición de la calidad de software.

Con la finalidad de definir, lo más cabalmente posible a la calidad del software, se debe en todo momento analizar la misma, desde dos perspectivas diferentes pero interrelacionadas, como lo son: el punto de vista del desarrollador y la del usuario final (cliente y/o consumidor). Aunque generalmente la que otorga el valor de “calidad” (capacidad de satisfacción), es la última.

Debido a la evolución constante de la calidad del software, y para asegurar su correcto uso, se debe englobar y establecer siempre los siguientes tres enfoques:

1. *Un sistema de calidad del proceso*: el cual provea de la correcta gestión, planificación y definición de políticas de calidad. Refiriéndose a todas las actividades, tareas, entradas, salidas, procedimientos, entre otros; que se realizan de manera secuencial para desarrollar y mantener el software. Así mismo, es en este caso, donde se puede hablar de técnicas, modelos, normas y metodologías tradicionales o ágiles. Por lo tanto, el proceso de desarrollo de software define cómo se organiza, gestiona, mide, soporta y mejora el desarrollo, independientemente de las técnicas y métodos usados. En este punto, la calidad depende de ¿Cómo?, se desarrolla o ejecuta el proceso y cada subproceso. Una desviación en cualquiera de las fases significaría puntos menos en la calidad del producto de software y por consiguiente en el nivel de satisfacción del usuario (Ramírez, Op. Cit.).
2. *El aseguramiento de la calidad (QA)*: se refiere a asegurar la calidad de los resultados de cada una de las fases del ciclo de vida de desarrollo del software y con esto, garantizar la calidad del producto final. Para cumplir con este aseguramiento la organización deberá definir estándares y establecer procedimientos contra los cuales se pueda comparar, lo alcanzado durante cada una de ellas con bucles de realimentación de información. Por ello el QA, abarca la revisión, auditoría e información sobre la adherencia de los procesos y productos a los estándares y procedimientos establecidos por una organización. Se debe entender como “producto” todos los entregables creados durante el proceso del desarrollo de software, incluyendo especificaciones, planes, procedimientos, código y documentación relacionada (Ramírez, Op. Cit.).
3. *Prácticas de control de calidad (QC)*: es el conjunto de procedimientos, mecanismos, acciones y herramientas para determinar si un producto de software cumple con sus metas de calidad (requisitos solicitados por el cliente y/o usuario). También se refiere a la capacidad del software para desempeñarse en escenarios impredecibles y mantener una tasa de defectos relativamente baja. Se debe hacer uso de técnicas de verificación y validación del software tales como la revisión e inspecciones de los productos de software que previenen la inserción de errores en el sistema. Tales como las pruebas de software que permiten controlar que el producto no llegue al cliente con problemas (Ramírez, Op. Cit.).

No obstante, se pueden establecer como las mejores definiciones reconocidas sobre la calidad del software, siendo estas las siguientes:

- La calidad de software es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente (Pressman, 2010).
- Según el modelo de la norma *ISO / IEC 25000*: Establece que la calidad del producto de software está compuesta de características de calidad, las cuales a su vez se componen de subcaracterísticas. Así mismo, establece que las medidas de calidad de software (*Software Quality Measures*) indican las características y subcaracterísticas de calidad del producto de software. Siendo estas, el grado en que el producto de software satisface las necesidades expresadas o implícitas, cuando se usa bajo condiciones determinadas.
- Según el modelo de la norma *ISO / IEC 25010*: Define a la calidad del producto de software como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor.
- Según la *IEEE Std. 610-1990*: Define a la calidad como el grado con el que un sistema, componente o proceso cumple con los requisitos especificados y las necesidades o expectativas del cliente o usuario.
- Según el *SEI*: La calidad de un sistema o de un producto está muy influenciada por la calidad del proceso empleado para desarrollarlo y para mantenerlo.

Con base en lo anterior, se ha formulado una propia definición:

“La CALIDAD DEL SOFTWARE es un grupo de características o atributos, que se desean medir y controlar, cuyas cualidades (valores) constituyen un conjunto de propiedades relacionadas a un producto de software (sistema), definidas propiamente por el usuario final (cliente y/o consumidor) como requisitos funcionales, de rendimiento, de fiabilidad y de seguridad. Cuya percepción de satisfacción (cumplimiento total de expectativas con el producto de software final), tanto de sus necesidades y/o perspectivas establecidas, otorgando así, un valor de supremacía y/o excelencia al mismo.

La calidad del software involucra el proceso de desarrollo (planificación, medida y seguimiento de las actividades llevadas a cabo), el cual tiene que hacer uso de un sistema de gestión de calidad que proporcionara confianza, al cliente de que se cumplirán sus requisitos con claridad, asegurando la misma con unas acciones de medida (control) de calidad, con un único y determinado propósito satisfacer plenamente al cliente”.

Es de vital importancia aclarar, lo que con mucha frecuencia las organizaciones confunden, la calidad del proceso vs. la calidad del producto, lo que genera falsas expectativas durante el desarrollo. Siendo esta última, la suma de todas las características de un producto que determina su capacidad para satisfacer las necesidades (requerimientos) de los clientes, consumidores y/o usuarios. Cuando cada producto de software tiene distintos requisitos de calidad, objetivos y contexto en el que se va a usar. Debido a todos estos, es que se vuelve interesante medir y mejorar unas características más que otras.

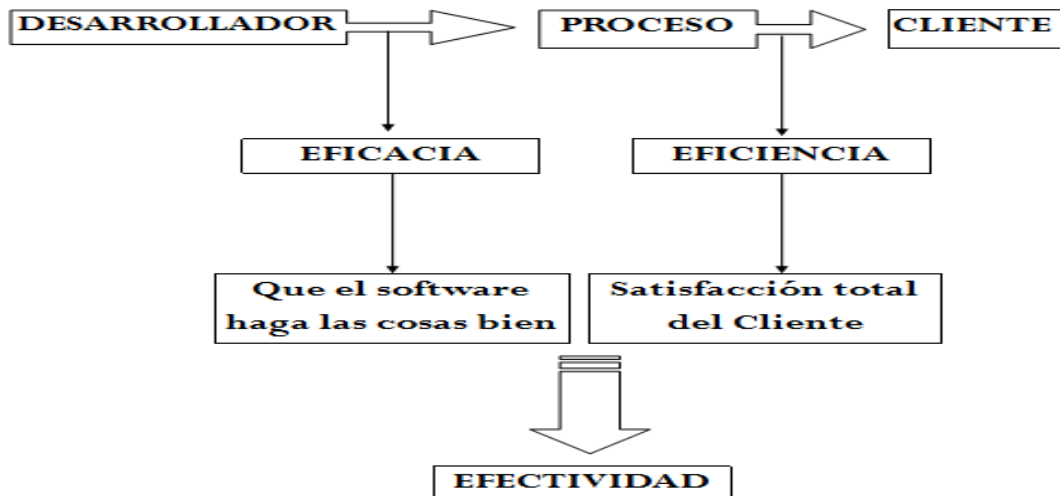
2.3.3.1. Elementos de la calidad del software.

Para lograr un mejor desarrollo de calidad en el software, se comienza identificando los elementos que intervienen, para su determinación dentro del mismo. Siendo los siguientes por orden de importancia:

1. Personal de desarrollo (desarrollador).
2. Proceso de desarrollo.
3. Cliente.
4. Técnicas, modelos y metodologías.
5. Normas y Estándares.
6. Documentación.

La Figura 3., ejemplifica un panorama en lo general sobre el proceso de jerarquización de los elementos de la calidad en el desarrollo de un producto de software anterior.

Figura 3. Proceso de determinación de la calidad en el desarrollo de software.

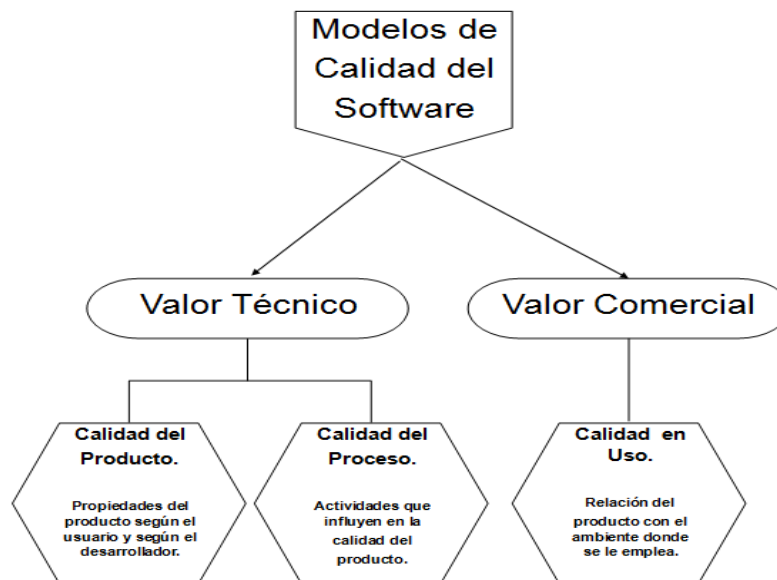


Fuente: Roberto Tenezaca, 2015.

2.3.3.2. Modelos de calidad de software.

Para usos prácticos se puede definir a la calidad de software en tres tipos de modelos, clasificándolos en dos tipos de valores, como lo ilustra la Figura 4., a continuación:

Figura 4. Modelos de Calidad del Software.



Fuente: Juan Manuel Fernández Peña, 2011.

Tomando en cuenta los dos valores, anteriores, se puede definir que los factores que pudiesen afectar la calidad, al producto de software, se les nombrara en lo sucesivo internos (Técnicos) y a los del cliente externos (Comercial). No obstante, para valorar adecuadamente la calidad del software, generalmente solo importan los factores externos, mismos que se obtienen correccionalmente de los internos.

- a) **Interna:** es medible a partir de las características intrínsecas del propio producto de software (por ejemplo, el código fuente).
- b) **Externa y/o uso:** es medible a partir del comportamiento y utilización efectiva del producto de software (por ejemplo, como los que se pueden medir durante una prueba o producción), por parte del usuario.

2.3.3.3. Factores y características que determinan la calidad del software.

Dentro de las características y funcionalidades que determinan la calidad de software, sin importar el tipo de proyecto de desarrollo, ilustradas en la Tabla 3., y se podría mencionar son las más trascendentes y significativas, siendo las siguientes:

Tabla 4. Factores y características que determinan la calidad del software.

FACTORES Y CARACTERÍSTICAS QUE DETERMINAN LA CALIDAD DEL SOFTWARE			
Tipo de Característica	Factor	Valor que satisface	Descripción
Característica Operativa del producto de software.	CORRECCIÓN	¿Hace lo que se le pide?	Es el grado en que un producto de software satisface sus especificaciones y consigue los objetivos encomendados por el cliente.
	FIABILIDAD	¿Lo hace de forma fiable todo el tiempo?	Es el grado que se puede esperar de un producto de software lleve a cabo las operaciones específicas y con la precisión requerida.
	EFICIENCIA	¿Qué recursos de hardware y software se necesitan?	Es la cantidad de recursos de hardware y software necesita un producto de software para realizar las operaciones con los tiempos de respuesta adecuados.
	INTEGRIDAD	¿Se puede controlar su uso?	Es el grado con que puede controlarse el acceso al software o a los datos a personal no autorizado.
	FACILIDAD DE USO	¿Es fácil y cómodo de manejar?	Es el esfuerzo requerido para aprender el manejo de un producto de software, trabajar con él, introducir datos y conseguir resultados. Revisión del producto: capacidad para soportar cambios.
Característica de transición del producto, a la adaptabilidad a nuevos entornos.	FACILIDAD DE MANTENIMIENTO	¿Se pueden localizar fallos?	Es el esfuerzo requerido para localizar y reparar los errores, que se presenten.
	FLEXIBILIDAD	¿Se pueden añadir nuevas opciones?	Es el esfuerzo requerido para modificar un producto de software en funcionamiento.
	FACILIDAD DE PRUEBA	¿Se puede probar todas las opciones?	Es el esfuerzo requerido para probar un producto de software de forma que cumpla con lo especificado en los requisitos. Transición del producto: adaptabilidad a nuevos entornos.
Característica de transición del producto, adaptabilidad a nuevos entornos.	PORTABILIDAD	¿Se podrá usar en otro dispositivo?	Es el esfuerzo requerido para transferir un producto de software a otro hardware o sistema operativo.
	REUSABILIDAD	¿Se podrá utilizar alguna parte del software en otro producto?	Es el grado en que una parte de un producto de software puede utilizarse en otros productos. Interoperabilidad (¿Podrá comunicarse con otros productos de software o sistemas informáticos?). Es el esfuerzo necesario para comunicar el producto de software con otros productos o sistemas informáticos.
	INTEROPERABILIDAD	¿Se podrá comunicar con otros productos o sistemas informáticos?	Es el esfuerzo necesario para comunicar el producto de software con otros productos o sistemas informáticos.

Fuente: (Tavera Moreno, 2012).

2.4. DEFINICION DE DISPOSITIVO MÓVIL.

2.4.1 Concepto de Dispositivo Móvil.

Un dispositivo móvil se puede definir como: un aparato de tamaño relativamente pequeño, con algunas capacidades de procesamiento, conexión permanente o intermitente a una red (Internet), con memoria limitada que ha sido diseñado específicamente para una función o funciones en específico, pero, no obstante, puede llevar a cabo muchas otras funciones más en paralelo (Dueñas, 2015).

En otras palabras, son pequeñas máquinas electrónicas que procesan datos, como lo hacen las computadoras de escritorio, algunos de estos pueden fácilmente conectarse a Internet y tienen memoria limitada. Con los dispositivos móviles, se pueden hacer muchas y diversas acciones como son: ver películas, realizar llamadas telefónicas y/o videollamadas, escuchar música, manejar una agenda, jugar videojuegos, por mencionar algunas.

2.4.2. Tipos de dispositivos móviles.

Los dispositivos móviles se dividen en : 1) Dispositivos móviles de datos limitados (Limited Data Mobile Device), caracterizados por ofrecer servicios de datos generalmente limitados a SMS y acceso WAP; 2) Dispositivos móviles de datos básicos (Basic Data Mobile Device), caracterizados por ofrecer servicios de acceso a emails, lista de direcciones, SMS y en algunos casos especiales un navegador web básico Y 3) Dispositivos móviles de datos mejorados (Enhanced Data Mobile Device), que se caracterizan por los servicios que ofrecen aplicaciones nativas como: Microsoft Office Mobile (Word, Excel, PowerPoint, por mencionar algunas) y aplicaciones corporativas usuales en versión móvil como: SAP. Portales de Intranet, entre otras y en algunas son diseñados exclusivamente para un sistema operativo específico, ejemplificado en la Figura 5.

Sin embargo, para la ejecución del prototipo de App que se creará, en el presente trabajo de tesis, solo se tomaran en cuenta el ultimo tipo de dispositivo móvil.

Figura 5. Enhanced Data Mobile Device.



Fuente: RDS Cloud Computing, 2014.

2.4.2.1. Sistemas operativos para dispositivos móviles actuales.

Los sistemas operativos (OS) fueron en algún tiempo únicamente hechos para computadoras de escritorio o computadoras portátiles (laptop, Netbook, por mencionar algunas), pero la evolución de los dispositivos móviles (smartphones, tabletas, reproductores mp3 y mp4, videoconsolas, entre otras), son muy diferentes y OS, tienden a ser más simples, orientados hacia la conectividad inalámbrica y necesidades más específicas, es decir, que con un dispositivo móvil, no es posible hacer todo lo que puedes hacer con una computadora, por obvias razones. Los OS móviles más utilizados actualmente son: Google Android, Apple iOS y Windows Phone mismos que se ejemplifican en la Tabla 5.

Tabla 5. Tipos de sistemas operativos Móviles

TIPO DE SISTEMAS OPERATIVOS MÓVILES			
Sistema Operativo	Número de Versión	Características	Tipo de Licencia
IOS	Versión 10	Swift es gratis y de código abierto, y está disponible para desarrolladores, educadores y estudiantes bajo la licencia de código abierto Apache 2.0. Proporciona archivos binarios para OS X y Linux que pueden compilar códigos para iOS, OS X, watchOS, tvOS y Linux. Además, para ayudar a que Swift sea un lenguaje aún más potente.	Código cerrado y propietario.
ANDROID	Versión 7.0 Nougat	Todos los desarrolladores de aplicaciones Android, tienen acceso total al código fuente usado en las aplicaciones base. Esto ha sido diseñado de esta forma, para que no se generen cientos de componentes de aplicaciones distintas, que respondan a la misma acción, dando la posibilidad de que los programas sean modificados o reemplazados por cualquier usuario sin tener que empezar a programar sus aplicaciones desde el principio. Android incluye en su base de datos un set de librerías C/C++, que son expuestas a todos los desarrolladores a través del framework de las aplicaciones Android System C library, librerías de medios, librerías de gráficos, 3D, SQLite, etc.	Código Abierto
WINDOWS PHONE	Versión Windows Phone 10	La plataforma Windows Phone soporta los lenguajes de programación C# y Visual Basic .NET. Esto se debe a que ambos son entendidos por el Framework .NET. Por la parte de diseño se utiliza el lenguaje Silverlight también conocido como XAML. En el caso de las aplicaciones de tipo video juegos, utilizan XNA para generar gráficos de tipo 2D y 3D e igualmente pueden hacer uso de Silverlight al mismo tiempo. Para programar, se necesita del ambiente de desarrollo Visual Studio 2010 Service Pack 1 en cualquiera de sus versiones.	Código cerrado y propietario.
FIREFOX OS	Versión 2.5.2	Si es un programador principiante busca escribir aplicaciones web, debería verificar el estado de implementación de la API. Las aplicaciones web son construidas usando tecnologías web estándar.	Código Abierto
BLACKBERRY OS	BlackBerry 10 OS, versión 10.3.2	La plataforma de BlackBerry y el desarrollo de aplicaciones móviles de recolección de datos con tecnologías de código abierto, incluyendo Java, PHP y MySQL. Las bases de la IU de BlackBerry y comunicaciones fueron presentadas, junto con un lado del servidor simple pero poderoso implementado en los básicos del código abierto PHP y MySQL.	Código cerrado y propietario.
UBUNTU TOUCH OS	Versión 8	Ubuntu Developer Tools Center (centro de herramientas para desarrolladores), un programa que servirá para poder poner una plataforma de desarrollo. Se ha centrado en los desarrolladores de Android, pero no se van a quedar ahí, piensan añadir soporte para Go, web, Dart y más. De momento Ubuntu Developer Tools Center está en un estado alfa, pero permite instalar la última versión de Android Studio con el último Android SDK y sus dependencias, activar la arquitectura múltiple en máquinas de 64bits e integrar Android Studio con el lanzador de Unity.	Código Abierto
TIZEN OS	Versión Tizen 2	Comprobar que el sistema de desarrollo cumple los requisitos especificados en los Requisitos del sistema de Android. Y Configura tu entorno Java (Para desarrollar aplicaciones de Android en Java, necesitas lo siguiente: Java Development Kit (JDK), proporciona las herramientas necesarias para crear una aplicación Java., Java Runtime Environment (JRE) permite ejecutar aplicaciones Java en tu ordenador).	Código cerrado y propietario.

Fuente: Elaboración propia, 2017.

2.4.2.2. Concepto de App para dispositivos móviles.

Las primeras Apps, datan de finales de los 90's estas eran los que se conocen como: agenda, *arcade games*, editores de *ringtone*, por mencionar algunas, cumplían funciones muy elementales y su diseño era bastante simple. La evolución de las Apps, se dio rápidamente gracias a las innovaciones en tecnología WAO y la transmisión de data (EDGE) esto vino acompañado de un desarrollo muy fuerte de los celulares y/o dispositivos móviles. El boom de las Apps se da dice la historia, cuando Apple Inc. lanza el iPhone y junto a él llegan muchas más propuestas de smartphones, entre ellas Android y empiezan las Apps de juegos, noticias, diseño, arte, fotografía, medicina y todo lo que se necesita para la vida diaria y que puede estar en tus manos (Leapp, 2012).

En este preciso momento, si alguien está determinado a utilizar un dispositivo móvil cualquiera que este sea, ciertamente usa unos programas informáticos llamados Apps, para realizar un sinnúmero de tareas, por ejemplo: obtener una geolocalización, acceder a noticias, leer libros, obtener datos climatológicos, por mencionar algunos.

Con base, en lo anterior y a la experiencia en el uso de las mismas, se ha formulado una definición propia, de aplicaciones móviles:

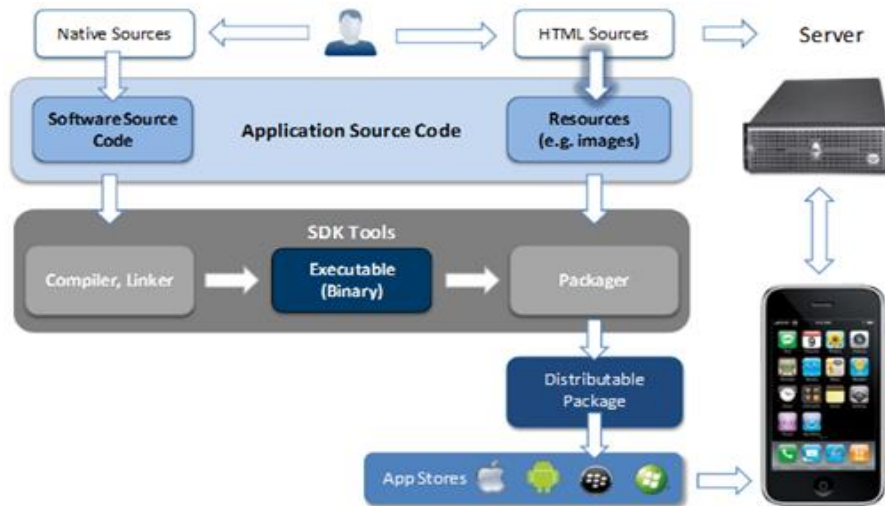
“Una APP, es un software que se descarga e instala en dispositivo móvil. Siempre se accede a ella desde el mismo dispositivo y cuyo objetivo será facilitar (al usuario) una labor y/o tarea concreta; así mismo asistirlo en las diversas gestiones y operaciones del día a día, ya sea de carácter profesional, personal, de ocio y/o entretenimiento. Ocupando generalmente, muy poco espacio de memoria, del mismo dispositivo móvil y se caracterizan por su dinamismo y practicidad. En algunos casos, estas se pueden usar en segundo plano, aun cuando se esté ejecutando algún otro proceso menor o mayormente pesado a la par”.

2.4.2.2.1. Características de las Apps.

Se pueden definir las siguientes características como esenciales, en una App útil, así mismo, se puede ejemplificar gráficamente en la Figura 6., respectivamente:

1. Facilidad de instalación.
2. Simplicidad de uso.
3. Ligereza (suelen ser programas ligeros con funcionalidades muy concretas).
4. Generalmente son gratuitas y cuando suelen ser de paga el costo es muy bajo (IEMD Marketing Digital, 2016).

Figura 6. Ejemplo de las características de una aplicación móvil.



Fuente: Geospatial Training, 2013.

2.4.2.2.2. Clasificación de las Apps.

1) APLICACIONES NATIVAS

Las App nativas, ilustradas en la Figura 7., se denominan así porque se desarrollan en el lenguaje nativo del propio terminal (dispositivo móvil). Dependiendo de la plataforma, en la que se desea desarrollar la App, se utiliza el lenguaje específico para cada OS, definido desde el fabricante. Los lenguajes de programación específicos para la terminal serán por lo tanto los siguientes:

- **iOS:** Objective C, Swift.
- **Android:** Java
- **Windows:** C# y Visual Basic .NET.
- **BlackBerry 10:** C y C++

Obviamente todo depende del nivel y experiencia del equipo de desarrollo y que el código resultante sea el correcto, pero en principio, una app nativa es la opción cuyo resultado es el más robusto y fluido; ya que se desarrolla directamente para integrarse en el OS. Sí la app surge de una buena idea y un diseño bien trabajado a todos los niveles, la experiencia del usuario será completa ya que su funcionamiento, rendimiento y respuesta será el más inmediato de todas las opciones de desarrollo, incluso en los diseños más complejos y personalizados, además de mostrar las ventajas ilustradas en la Tabla 6 (Appio, 2013).

Figura 7. Ejemplo de aplicación nativa.



Fuente: Jacobo Varela, 2015 y OpenWebinars 2016.

Tabla 6. Ventajas y desventajas de las aplicaciones nativas.

VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none"> - Pueden utilizar todos los recursos y elementos del sistema y del hardware (cámara, GPS, etc.). - La experiencia del usuario es más completa - No necesitan conexión a Internet para funcionar - Se distribuyen a través de las App store de los fabricantes. - Actualización constante. 	<ul style="list-style-type: none"> - Se necesitan desarrolladores con conocimientos específicos. - Sólo funcionan en aquellos dispositivos que posean el sistema operativo para el que se desarrolló. - Los fabricantes tienen que aprobarla para hacerla accesible al público general.

Fuente: Avante, 2015.

2) APLICACIONES HÍBRIDAS

Generalmente consisten en Apps que contienen en su interior el navegador web del dispositivo. Para su desarrollo se utilizan *frameworks* de desarrollo basados en lenguajes de programación web (HTML, CSS y JS), como se ilustra en la Figura 8. En este tipo de Apps el nivel de integración con el OS dependerá del *frameworks* de desarrollo utilizado y que tan abierto sea el mismo, teniendo cada uno de ellos sus ventajas e inconvenientes. Actualmente con esta opción, se tendrá bastante acceso al hardware del dispositivo e incluso en algunos casos a las librerías propias del OS, pero el inconveniente es que, aunque de momento no se ha conseguido igualar la respuesta y la experiencia de usuario de una app nativa, hay que reconocer que va camino a hacerlo.

Sin embargo, han ido evolucionando a desarrollos más complejos en los que ciertas funcionalidades se ejecutan como web y en otras en nativo, las Apps híbridas son una opción muy económica y muy interesante para llegar al mayor número de usuarios repartidos en las diferentes plataformas y dispositivos, aunque por el momento sus limitaciones son claras y ofrecen las ventajas ejemplificadas en la Tabla 7 (Appio, Op.Cit.).

Figura 8. Ejemplo de aplicación híbrida.



Fuente: Jacobo Varela, 2015 y *OpenWebinars* 2016.

Tabla 7. Ventajas y desventajas de una aplicación híbrida.

VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none">-La mayoría son de uso gratuito.- El coste de desarrollo es menor que el de una aplicación nativa.- Pueden ejecutarse en múltiples dispositivos.- Pueden ser distribuidas masivamente.	<ul style="list-style-type: none">-El desarrollo de estas aplicaciones es algo complicado.

Fuente: Avante, 2015.

3) APLICACIONES WEB

Una app web es precisamente, una web a la que se accede a través de una URL en el navegador del dispositivo (Chrome, Safari, Opera, por mencionar algunos), ilustrada en la Figura 9., y se adapta al formato de la pantalla del dispositivo, para que tenga aspecto de navegación app. Los navegadores de los dispositivos permiten crear un acceso directo en el escritorio de esta web, así que será la manera de “instalarla” (sí se puede llamar así) en el dispositivo. A nivel de lenguajes de programación pues efectivamente es lo que supone, usar lenguajes de programación web, ofreciendo las ventajas definidas en la Tabla 7 (HTML, CSS, JavaScript), (Appio, Op.Cit.).

Figura 9. Ejemplo de aplicación web.



Fuente: Jacobo Varela, 2015 y OpenWebinars 2016.

Tabla 8. Ventajas y desventajas de una aplicación web.

VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none"> -Se pueden ejecutar en multitud de dispositivos. - El desarrollo es más sencillo al utilizar tecnologías muy conocidas como HTML. - Podemos encontrarlas en cualquier buscador. - El usuario siempre dispone de la última versión. - Son más baratas de desarrollar que las nativas. - No requieren la aprobación de ningún fabricante. 	<ul style="list-style-type: none"> -No pueden publicarse en las stores para su distribución masiva. - No aprovechan el hardware y recursos del dispositivo de un modo óptimo. - Requieren conexión a Internet. - Precisan de un mayor esfuerzo en promoción y visibilidad.

Fuente: Avante, 2015.

2.4.3. Tipos de usuarios

- **Tipos de Usuarios de Dispositivos Móviles**

La relación entre el usuario y su dispositivo se puede agrupar de la siguiente manera, así mismo, se ejemplifica de forma gráfica en la Figura 10, respectivamente:

1. Básicos: Su uso se enfoca en la comunicación.
2. Inquietos: Los dispositivos como son una fuente de distracción.
3. Sociales: Buscan tener estatus y estar a las modas a través de sus dispositivos.
4. Prácticos: Valoran la practicidad de sus dispositivos móviles.
5. Expertos: Sus dispositivos son su conexión con el mundo (Puon, 2014).

Figura 10. Tipos de Usuarios de Dispositivos Móviles.



Fuente: Liz Puon, 2014.

Actualmente debido a las grandes posibilidades y capacidades de los dispositivos móviles, ya se pueden definir una serie de consumidores basado en un estudio de usos y hábitos en México.

- **Tipos de usuarios de Aplicaciones Móviles**

Con base, en las descargas de aplicaciones móviles tiene mucha influencia la nacionalidad del mismo. En un estudio realizado por NOKIA, esta va más allá al segmentar a los usuarios en 6 tipos de arquetipos, llamados Apli-tipo y los divide en estas categorías a los mismos, según los tipos de Apps, que descargan en sus dispositivos.

1. *AppTusiastas*: Son consumidores energéticos y apasionados, son personas muy sociables que disfrutan de las tendencias. Curiosos y espontáneos, disfrutan de estar involucrados en muchos tipos diferentes de actividades. Este grupo de consumidores está formado generalmente por adolescentes de entre 14 y 18 años de edad.
2. *AppCéntricos*: Son consumidores trabajadores, energéticos y productivos. Suelen ser “frikis” tecnológicos y suelen buscar las últimas novedades en sus teléfonos que les ayuden en sus tareas. Este grupo de consumidores está formado generalmente por adultos de entre 25 y 40 años, de un nivel socioeconómico alto.
3. *AppConectados*: Son consumidores activos que disfrutan desde búsquedas técnicas, relativas al disfrute de actividades al aire libre y deportivas. Son sociables, tienen muchos amigos y muchas veces buscan nuevas experiencias y aventuras, disfrutando de cualquier desafío. Este grupo de consumidores está formado generalmente por adolescentes y adultos de hasta 40 años.
4. *AppCreadores*: Son consumidores creativas e imaginativas. Son personas que disfrutan de contar historias, son cálidos y entusiastas no solo dependen de las capacidades creativas de su dispositivo, pero a menudo prefieren el uso de redes sociales. Este grupo de consumidores está formado generalmente por adultos de entre 25 y 34 años y en algunos casos extremos se vuelven “bajadores compulsivos”.

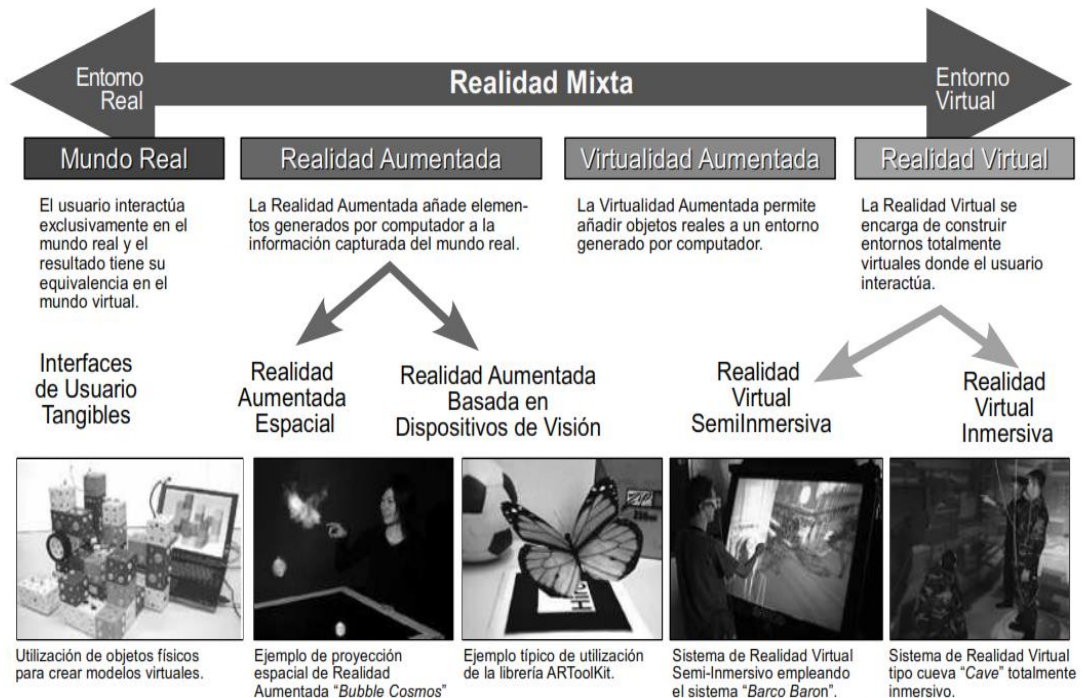
5. *AppConectores*: Son consumidores afables, confiados y curiosos. Tienen una personalidad estimulante y motivadora con una extraordinaria habilidad para hacer amigos en línea y/o en el mundo real. Cuentan con una colección de aplicaciones que han utilizado una y otra vez, a menudo llevando descargas relativas a la diversión en su dispositivo, así como aplicaciones que pueden ser útiles a los demás. Este grupo de consumidores está formado generalmente por adultos de entre 35 y 40 años y suelen tener un alto poder adquisitivo.
6. *Apprendices*: El consumidor aprendiz es sensible, bondadoso e independiente. Son curiosos e inteligentes con un círculo sólido de amigos y un selecto grupo de búsquedas que les apasiona. Los aprendices están interesados en la descarga de aplicaciones, pero muchas veces es tanta su emoción que terminan por no saber dónde empezar (Soria, 2011).

2.5. REALIDAD AUMENTADA

2.5.1. Concepto de realidad aumentada (AR).

La AR, trata de perfeccionar el mundo real, suponiendo o añadiendo capas de información. Mediante elementos creados de manera digital, sobre el entorno que rodea al usuario. Esta es la principal diferencia entre la realidad aumentada y virtual, se utiliza el mundo real como parte de la experiencia, e incluso se interactúa con él. Además, para poder disfrutar de la AR, el usuario no necesita ningún tipo de dispositivo (artefacto), que aisle completamente sus sentidos del entorno en el que se encuentra, si no que los dispositivos que se utilizan permiten combinar, simultáneamente ambas capas de información, con lo que se ve en tiempo real, como lo ilustra la Figura 11 (Javier Sáez, González, Soto, & Carnero, 2017).

Figura 11. Taxonomía de realidad mixta según Milgram y Kishino.



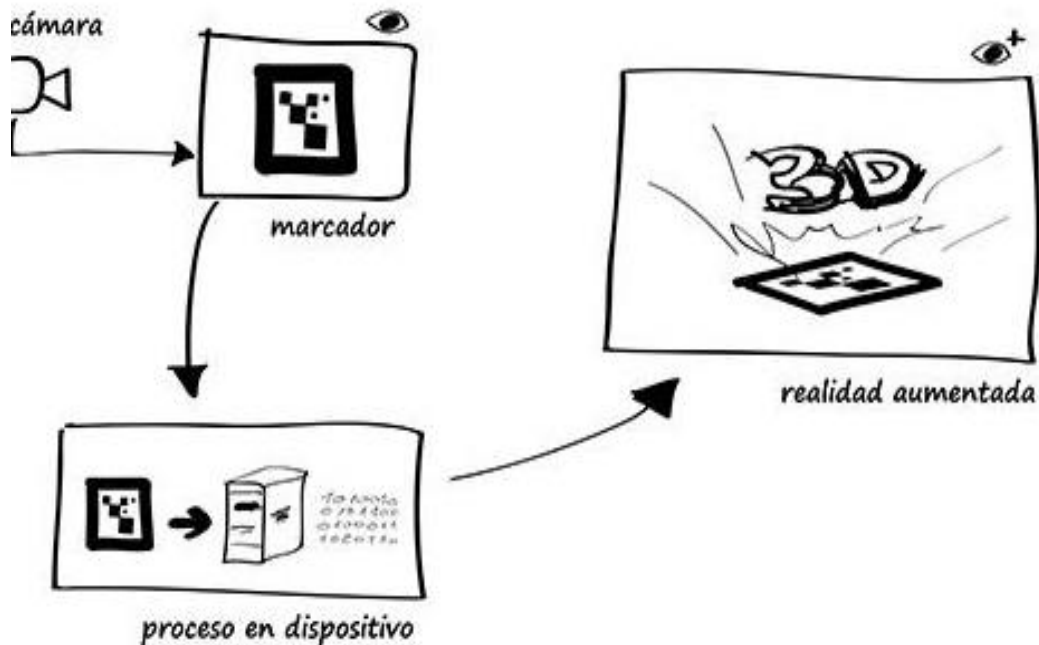
Fuente: (González Morcillo, Vallejo Fernández , Albusac Jiménez, & Castro Sánchez , 2012).

2.5.2. Características Generales de la realidad aumentada (AR).

Según Ronald T. Azuma, un sistema de AR debe cumplir las siguientes características, como lo muestra gráficamente la Figura 12.:

1. *Combina el mundo real y el mundo virtual.* El sistema incorpora información sintética a las imágenes percibidas del mundo real.
2. *Interactivo en tiempo real.* Así, los efectos especiales de películas que integran perfectamente imágenes 3D fotorrealistas con imagen real no se considera realidad aumentada porque no son calculadas de forma interactiva.
3. *Alineación 3D (registro tridimensional).* La información del mundo virtual debe ser tridimensional y debe estar correctamente alineada con la imagen del mundo real. Así, estrictamente hablando las aplicaciones que superponen capas graficas 2D y /o 3D, sobre la imagen del mundo real no son consideradas de realidad aumentada (González y otros, Op. Cit.).

Figura 12. Proceso de la realidad aumentada (AR), con marcador (*target*).



Fuente: Oreka interactive, 2015.

La combinación de estas tres características hace que la *AR*, sea muy interesante para el usuario, ya que complementa y mejora su visión e interacción del mundo real, con información que puede resultar extremadamente útil, a la hora de realizar ciertas tareas. De hecho, la *AR* es considerada como una forma de amplificación de la inteligencia que emplea el computador para facilitar el trabajo del usuario.

2.5.3. Elementos de la realidad aumentada (AR).

Para poder conseguir esa superposición de los elementos virtuales de un entorno físico, en un proceso de realidad aumentada, debe estar formado por los siguientes elementos:

1. *Cámara*. Es el dispositivo que capta la imagen del mundo real. Puede ser la webcam para computadora o bien la cámara de un dispositivo móvil.
2. *Procesador*. Es el elemento de hardware que combina la imagen con la información que debe sobreponer.
3. *Software*. Es el programa informático específico que gestiona el proceso.
4. *Pantalla*. En ella se muestran combinados los elementos reales y virtuales.

5. *Activador y/o Marcador.* Es un elemento del mundo real que el software utiliza para reconocer el entorno físico y seleccionar la información virtual asociada que se debe añadir. Puede ser un código QR (del inglés *Quick Response Code* “código de respuesta rápida”), un marcador, una imagen u objeto, o bien la señal GPS (del inglés, *Global Positioning System* “Sistema de posicionamiento global”), enviada por el dispositivo y que será donde se añadirá el modelo tridimensional virtual.
6. *Imágenes u objetos.* Es la imagen, objeto o animación, que se añadirá a la capa correspondiente con información virtual.
7. *Conexión a Internet.* Se utiliza para enviar la información del entorno real al servidor remoto y recuperar la información virtual asociada que se superpone (Posada, 2014).

La Figura 13., representa dos ejemplos del uso de la realidad aumentada más comunes en la actualidad: por un lado, un Smartphone que incorpora una aplicación de realidad aumentada y por otro, el caso de un PC con una aplicación de realidad aumentada en el que la activación se produce por un marcador impreso.

Figura 13. Ingredientes básicos de la realidad aumentada, ejemplos de uso en Smartphone y en pc con códigos BIDI.



Fuente: Fundación Telefónica, 2011.

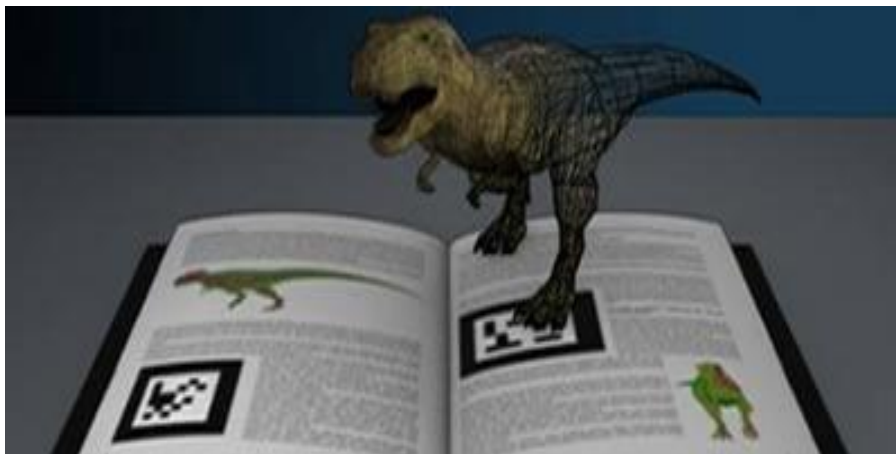
2.5.4. Tipos de tecnologías de realidad aumentada (AR)

Se puede considerar que existen actualmente, dos tipos de AR:

- **Realidad aumentada basada en marcadores.**

En este tipo de AR, se utiliza un marcador, como se ilustra en la Figura 14., que una vez reconocido, por el software de aplicación de la AR, añade el objeto en realidad virtual, que se ha programado. Este tipo ha sido el primero de usarse. Sin embargo, en este momento, se puede utilizar cualquier imagen plana, 2D, 3D e incluso 4K, como marcador (Tarjet), y se están teniendo los primeros resultados, para poder utilizar cualquier cuerpo cilíndrico (por ejemplo, una lata de refresco), así como el reconocimiento de patrones (formas), colores y/o distancias (medidas).

Figura 14. Ejemplo de realidad aumentada con marcadores.



Fuente: (Pinto, López de Toledo, & Muñoz , 2015).

- **Realidad aumentada sin marcadores.**

En este tipo de AR, se utiliza la geolocalización, como lo muestra la Figura 15., información que se obtiene del *GPS* del dispositivo móvil, el software de la App de AR sabe perfectamente, qué información y en qué momento, se debe añadir a la

imagen virtual, ya se de tipo plana, texto, 2D, 3D 4K, botón interactivo, video y/o audio a una imagen real.

Figura 15. Ejemplo de realidad aumentada sin marcadores, videojuego “Pokemon Go”.



Fuente: Niantic, Inc. / Nintendo / Creatures Inc. / GAME FREAK inc, 2017.

2.5.5. Herramientas para desarrollo de *realidad aumentada (AR)*.

2.5.5.1. UNITY 5.3.5 f1 y 5.4.0 f3

Es un motor de desarrollo para videojuegos multiplataforma y contenidos 3D, con la característica, de integridad completa y ofrece innumerables funcionalidades facilitando su creación. Con *UNITY*, cuyo logotipo y ambiente de trabajo se ilustra en la Figura 16., se puede acceder a plataformas móviles de realidad virtual (*VR*, por sus siglas en inglés), en *Smartphones*, navegadores Web, Xbox 360, Wii U, consola PS3 y aparatos de TV, entre otros; para los que se puede, desarrollar juegos que van desde la guía de operaciones de manejo de materiales (*MMOG*, por sus siglas en inglés), videojuego de disparos (*shooters*), hasta juego de roles, cuyos requerimientos de instalación se muestran en la Tabla 9.

Figura 16. Ejemplo de motor de desarrollo UNITY.



Fuente: Alberto Vela, 2011 y bestwebsoft.com, 2014.

UNITY es una solución flexible, rápida y de alta gama, intuitivo y altamente personalizable. Con poder de renderizado y sombreado (*shading*, en el argot de esta herramienta) de base física altamente optimizado, se pueden crear juegos hermosos, con suma rapidez.

2.5.5.1.1. Requerimientos de sistema de instalación del motor UNITY.

Tabla 9. Requisitos de sistema para versión de UNITY.

REQUISITOS DE SISTEMA PARA VERSIÓN DE UNITY			
PARA DESARROLLO			
Sistema Operativo (OS)		Unidad de Procesamiento Gráfico (GPU)	
<ul style="list-style-type: none"> Windows 7 SP1+, 8 y 10. Mac os x 10.8+ 		Tarjeta gráfica con DX9 (modelo de shader 3.0) o DX11 con capacidades de funciones de nivel 9.3.	
REQUISITOS ADICIONALES PARA EL DESARROLLO DE PLATAFORMAS			
iOS	Android	Windows Store	WebGL
Computadora MAC con OS versión OSX 10.9.4, como mínimo y Xcode 7.0 o superior.	Android SDK y Java Development Kit (JDK).	Windows 8.1 (64 Bits), Visual Studio y plataforma SDK, correspondiente: Windows 8.1 /Windows Phone 8.1: Visual Studio 2013 o versión posterior y Windows 8.1 SDK. UWP: Visual Studio 2015 o versión posterior y Windows 10 SDK. El scripting backend IL2CPP también requiere la instalación de la función de compilador C++ con Visual Studio.	Mac OS X 10.8+ o Windows 7 sp1+ (sólo editor de 64 bits).
PARA EJECUTAR JUEGOS DE UNITY			
Escritorio	iOS	Android	WebGL
OS: Windows XP SP2+, Mac OSX 10.8+, Ubuntu 12.04+, StreamOS+. GPU: DX9 (modelo de shader 3.0) o DX11 con capacidades de funciones de nivel 9.3. <ul style="list-style-type: none"> CPU: compatible con el conjunto de instrucciones SSE2. 	El reproductor de iOS, requiere iOS 7.0 o superior (antes requería 6.0)	OS 4.1 o posterior; ARMv7 (Cortex) CPU con soporte para NEON o CPU Atom; OpenGL ES 2.0 o posterior.	Cualquier versión de escritorio reciente de Firefox, Chrome, Edge o Safari.
Windows Phone y Windows Store Apps			
8.1 o posterior			

Fuente: Unity3d.com, 2017.

2.5.5.2. VUFORIA SDK v6.2.x

VUFORIA es un SDK (Kit de desarrollo de software), permite construir aplicaciones basadas en la AR, una App desarrollada con *VUFORIA* utiliza la pantalla de un dispositivo cualquiera, como un "lente mágico" en donde se entrelazan elementos del mundo real con elementos virtuales como: modelos, bloque de textos, imágenes, por mencionar algunos, cuyo involucramiento de desarrollo, se muestra en la Figura 17.

Figura 17. Envolvimiento de desarrollo VUFORIA.



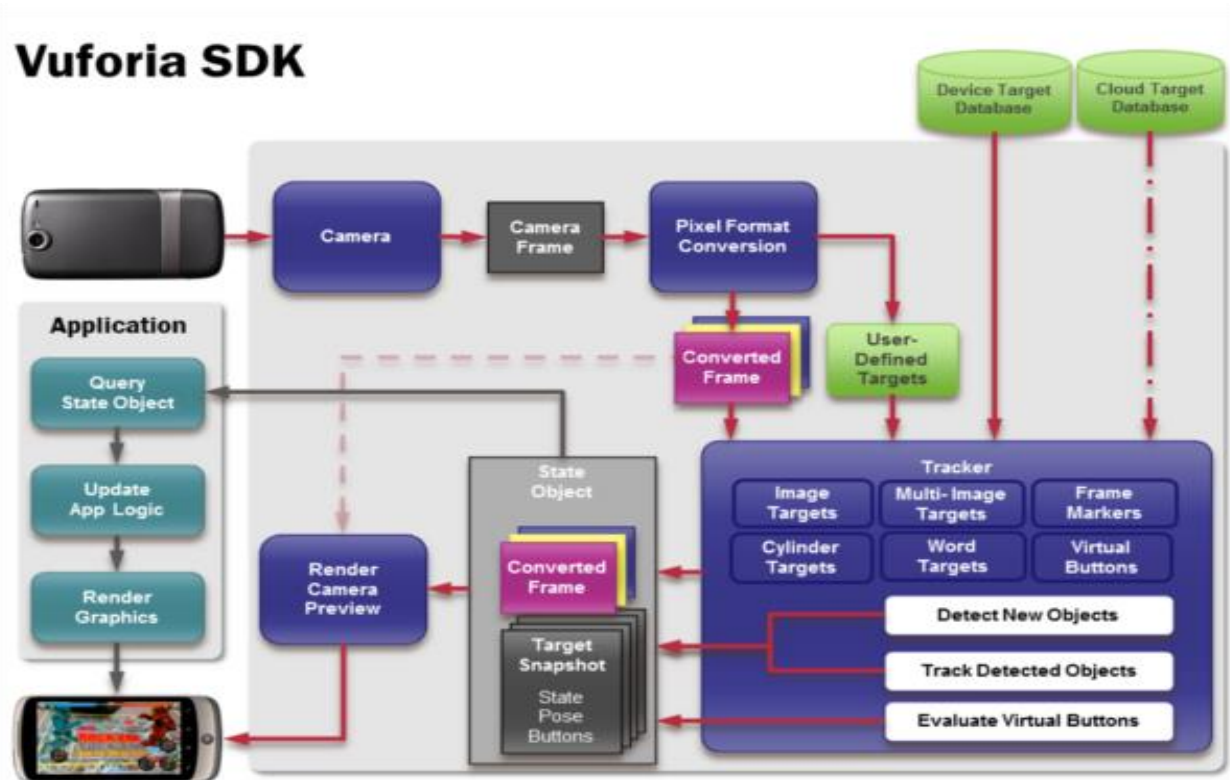
Fuente: Au-Duong Tin Yung, Luk Hoi Kit, Ng Yui Wang and Jeff Tang, 2014.

2.5.5.2.1. Arquitectura de Vuforia

Una App desarrollada con *VUFORIA* está compuesta de los siguientes elementos y se ciñe a la arquitectura ilustrada a detalle en la Figura 18.:

- *Cámara*: asegura que la imagen sea captada y procesada por el *Tracker*.
- *Base de datos*: la base de datos del dispositivo es creada utilizando la gestión de objetivos (*Target Manage*); ya sea la base de datos local o la base de datos en la nube, almacena una colección de objetivos (*Targets*), para ser reconocidos por el *Tracker*.
- *Target*: Son utilizadas por el rastreador (*Tracker*), para reconocer un objeto del mundo real; los *Targets* pueden ser de diferentes tipos; entre los principales tenemos:
 - *Image Targets*: son imágenes; tales como: fotos, páginas de revistas, cubierta de libros, poster, tarjetas, entre otras.
 - *Word Targets*: Elementos textuales que representan palabras simples o compuestas, libros, revistas, por mencionar algunas. Hay dos modos de reconocimiento posible: la palabra entera o por caracteres.
- *Tracker*: Analiza la imagen de la cámara y detecta objetos del mundo real a través del marco (*Frame*), de la cámara con el fin de encontrar coincidencias en la base de datos (Cruz, 2014).

Figura 18. Diagrama de flujo sobre la arquitectura de datos del SDK Vuforia en una aplicación móvil.



Fuente: Akanksha, 2014.

Se puede observar en la Figura 18., el proceso el cual comienza cuando el dispositivo móvil, capta una escena (un video en vivo) tomada a través de la cámara, la SDK de VUFORIA crea un *frame* (una imagen particular dentro de una sucesión de imágenes) de la escena capturada y convierte la imagen capturada por la cámara, a una diferente resolución para ser correctamente tratada por el *tracker*, VUFORIA SDK analiza la imagen a través del *tracker* y busca coincidencias en la base de datos, la cual está compuesta por targets, luego la aplicación hace "algo"; este "algo" es renderizar algún contenido virtual (imágenes, videos, modelos, etc.) en la pantalla del dispositivo, y así crear una realidad mixta con elementos virtuales combinados, conjuntamente con elementos reales, o lo que se conoce como AR (Cruz, Op.Cit.).

2.5.5.2.2. Requerimiento de sistema para instalación de Vuforia

Esta es una gran ventaja para la instalación de VUFORIA SDK, está disponible para las OS, más comunes en el mercado y que se describen en la Tabla 10., a continuación:

Tabla 10. Requerimientos del sistema Vuforia SDK.

REQUERIMIENTOS DEL SISTEMA VUFORIA SDK	
Tipo	Características
Plataformas de desarrollo	<ul style="list-style-type: none"> Windows Linux Mac
Plataformas móviles soportadas	<ul style="list-style-type: none"> Android ios
Corrección de errores / características	<ul style="list-style-type: none"> Todas: Vuforia ahora es compatible con el seguimiento de un número ilimitado de objetivos imagen de forma simultánea; Problema de rendimiento fija con VuMarks cuando se utiliza la "vía apariencia de" opción. Android 8.2.10: se ha solucionado un problema que causa una imagen de vista previa de cámara oscura en algunos dispositivos Nexus y LG; fijar una pérdida de memoria en Android al pasar la primera actividad durante la inicialización

Fuente: Cruz, Op. Cit.

2.5.5.3. Lenguajes de Programación e IDE

Los lenguajes de programación utilizados y el IDE de complemento para poder utilizar la AR en un dispositivo móvil Android se ilustran en la Tabla 11 y Tabla 12., respectivamente en lo sucesivo:

- Lenguaje de programación**

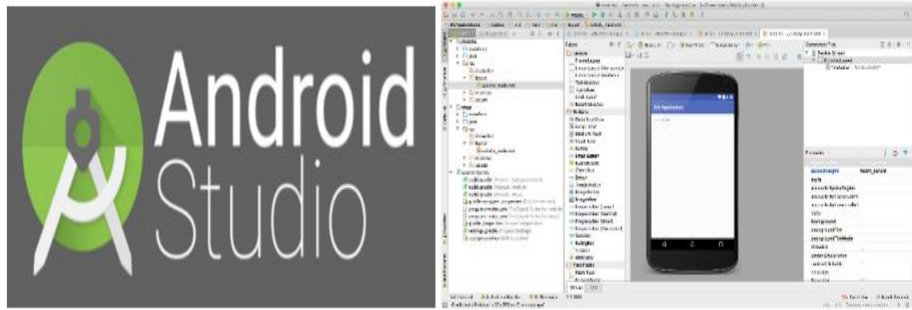
Tabla 11. Lenguajes de programación e IDE.

The image shows a screenshot of a document page with a blue header and footer. The main content area contains text and code snippets. The text appears to be technical documentation related to AR development on Android, mentioning various programming languages and IDEs. The code snippets are in a monospaced font, typical of programming code. The page is somewhat blurry and has a low resolution.

Fuente: Elaboración Propia, 2017.

- **ANDROID STUDIO Y SDK BUILD TOOLS V. 23.0.3**

Figura 19. Ejemplo de desarrollo y logo de Android Studio.



Fuente: Lorenzo Delli, 2016 y *DevelopersAndroid*, 2017.

Android Studio, cuyo logo está ilustrado en la Figura 19., proporciona las herramientas más rápidas para crear Apps en todas las clases de dispositivos Android. La edición de códigos de primer nivel, la depuración, las herramientas de rendimiento, un sistema de compilación flexible y un sistema instantáneo de compilación e implementación que permite concentrarse en la creación de aplicaciones únicas y de alta calidad.

Tabla 12. Requisitos de sistema para Android Studio.

REQUISITOS DEL SISTEMA PARA ANDROID STUDIO		
Windows	Mac	Linux
<ul style="list-style-type: none"> Microsoft® Windows® 7/8/10 (32 o 64 bits). 3 GB de memoria RAM como mínimo (se recomiendan 8), más 1 GB para el emulador de Android. 2 GB de espacio en disco disponible como mínimo (se recomiendan 4); 500 MB para el IDE + 1,5 GB para Android SDK y la imagen de sistema del emulador. Resolución de pantalla mínima de 1280 x 800. Para el emulador acelerado: Sistema operativo de 64 bits y procesador Intel® compatible con Intel® VT-x, Intel® EM64T (Intel® 64) y la funcionalidad Execute Disable (XD) Bit. 	<ul style="list-style-type: none"> Mac® OS X® 10.10 (Yosemite) o versiones posteriores, hasta la 10.12 (MacOS Sierra). 3 GB de memoria RAM como mínimo (se recomiendan 8), más 1 GB para el emulador de Android. 2 GB de espacio en disco disponible como mínimo (se recomiendan 4); 500 MB para el IDE + 1,5 GB para Android SDK y la imagen de sistema del emulador. Resolución de pantalla mínima de 1280 x 800. 	<ul style="list-style-type: none"> GNOME o KDE de escritorio. Pruebas realizadas en Ubuntu® 12.04, Precise Pangolin (distribución de 64 bits capaz de ejecutar aplicaciones de 32 bits). Distribución de 64 bits capaz de ejecutar aplicaciones de 32 bits. GNU C Library (glibc) 2.19 o versiones posteriores. 3 GB de memoria RAM como mínimo (se recomiendan 8), más 1 GB para el emulador de Android. 2 GB de espacio en disco disponible como mínimo (se recomiendan 4); 500 MB para el IDE + 1,5 GB para Android SDK y la imagen de sistema del emulador. Resolución de pantalla mínima de 1280 x 800. Para el emulador acelerado: Procesador Intel® compatible con Intel® VT-x, Intel® EM64T (Intel® 64) y la funcionalidad Execute Disable (XD) Bit, o procesador AMD compatible con AMD Virtualización™ (AMD-V™)

Fuente: (Android Studio Blog Developer, s.f.).

2.5.6. Ejemplos de Apps lingüísticas sustentadas en Realidad Aumentada (AR).

Si se tomaran en cuenta, diversos aspectos, encontrados en el vasto mundo de Apps, actualmente disponibles a nivel mundial. El sin número de Apps para aprender idiomas con o sin el uso de la AR es infinito. No obstante, se puede tomar en cuenta que tienen como objetivo principal el mismo, que el prototipo App Draw-Hánzǐ, del presente trabajo de tesis. Sin embargo, se pueden destacar como las más afines, las siguientes:

1. FluentU 1.3.8

FluentU es una App, que ofrece diversos cursos para aprender español, francés, chino mandarín, alemán, japonés e inglés. El objetivo principal de FluentU es mejorar el nivel del idioma que se desea aprender, mediante la inmersión al mismo, a través de videos musicales, avances de películas, noticias y charlas inspiradoras, cuya interfaz se ilustra en la Figura 20. FluentU está disponible para iOS 8 o posterior y es compatible con iPhone, iPad y iPod touch. FluentU tiene un precio de \$15.00 USD / al mes plan básico, \$29.99 USD / al mes Plan Plus o \$239.00 USD / al año Plan Plus (FluentFlix Limited, 2017).

Figura 20. Aplicación Móvil FluentU para aprender idiomas.

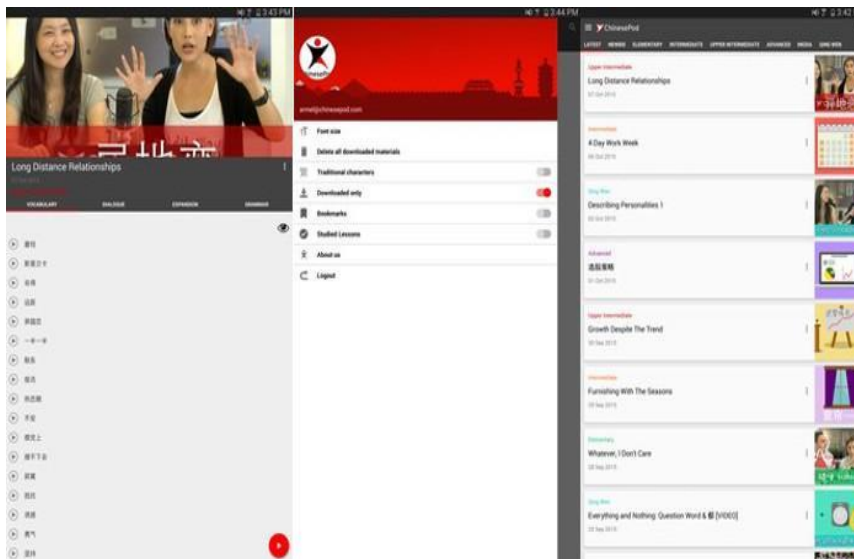


Fuente: *FluentFlix Limited, 2017.*

1. Chinese Pod

ChinesePod es una de las plataformas más prestigiosas y una de las más antiguas para aprender chino, recientemente ha desarrollado una App para sistemas operativos Android 4.1 o superior e iOS 7 o superior, que permite seguir un plan de estudios, en cualquier lugar, sin conexión en internet indispensablemente. La App ChinesePod, ilustrada en la Figura 21., ofrece todos los servicios de la plataforma, que incluye vídeos y podcasts organizados por niveles, así como materiales tales como ejercicios de revisión para lecciones y tarjetas con listas de vocabulario personalizadas de apoyo cuya biblioteca es de 4,000 lecciones aproximadamente. *ChinesePod* tiene un precio de \$14.00 USD / al mes por el Plan Básico y \$ 29.00 USD / al mes por el Plan Plus (*ChinesePod Limited, 2016*).

Figura 21. Podcast de *Chinese Pod* para aprender el idioma chino



Fuente: ChinesePod Ltd., 2017.

2. Chairman's Bao Chinese New

The Chairman's Bao es una App, ilustrada en la Figura 22., para aprender el idioma chino simplificado o tradicional, para estudiantes de un nivel avanzado, pero también puede ser un buen recurso para los principiantes, ya que publica alrededor de 1,600 noticias al año, tiene un diccionario interactivo (al leer una nota puede traducir [pronunciación "audio" y definición], de una palabra en tiempo real), se puede hacer

una base de datos de palabras desconocidas y crear grupos de palabras, sistema de tarjetas flash con una lista de vocabulario y expresiones idiomáticas en los artículos para no perder ningún detalle y escritura de caracteres con videos animados y herramienta de prueba de las mismas. The Chairman's Bao tiene un precio de \$10.00 USD / al mes o \$25.00, \$45.00 y \$80.00 USD / para 3, 6 y 12 meses, respectivamente (*The Chairman's Bao Ltd., 2017*).

Figura 22. Aplicación Chairman's Bao para aprender chino



Fuente: *The Chairman's Bao Ltd, 2017*.

3. Memrise

Memrise es una App, para aprender idiomas como: el francés, español (de México o de España), alemán, árabe, inglés, chino, japonés, coreano, turco, ruso, italiano, portugués (de Brasil o Portugal), noruego, danés, por mencionar algunos, el formato es una especie de videojuego, como lo muestra la Figura 23., en el cual todo lo que aprendes, te sirve para engañar a los agentes enemigos y/o para detectar otros agentes que te ayudaran en el camino. Está disponible para los sistemas operativos Android e iOS 9 o posterior y es compatible con iPhone, iPad y iPod touch. Memrise en la versión básica es gratuita, y la versión Memrise Pro tiene un precio de \$ 8.99 USD / al mes, \$18.99 USD / los tres meses, \$59.99 USD / por los 12 meses y \$179.99 USD por una suscripción indefinida (Memrise, 2017).

Figura 23. Aplicación móvil Memrise para aprender idiomas

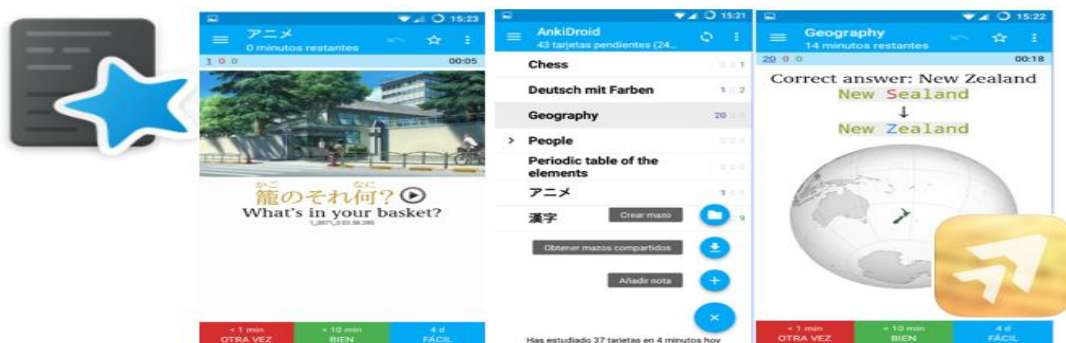


Fuente: Memrise, 2015.

4. AnkiDroid & AnkiApp

AnkiDroid & AnkiApp es la App, más popular para estudiar y/o memorizar, con el uso de tarjetas o aprender nuevas palabras. AnkiDroid & AnkiApp, ilustrada en la Figura 24., se pueden crear las cubiertas o personalizarlas, ya que se descargan gratuitamente desde la base de datos pública de la app, así como también se pueden crear conjuntos de tarjetas personales y/o públicas. Es una App que también existe para escritorio y es muy sencilla de sincronizar ambas entre sí, está disponible para los sistemas operativos de escritorio: Windows, Mac y Linux, así como los sistemas operativos móviles: Android 2.3.3. o superior, es compatible con tablets e iOS 8 o superior es compatible para iPhone, iPad y iPod touch. AnkiDroid es gratuito. Sin embargo, AnkiApp para iOS tiene un precio de \$24.99 USD /por mes (*AnkiDroid Open Source Team & Admium Corp., 2017*).

Figura 24. Aplicación móvil AnkiDroid para memorizar por medio de tarjetas



Fuente: *Android Open Source Team y Admium Corp., 2017.*

5. Hello Talk

Hello Talk es una App, muy similar a Whatsapp, como se muestra en la Figura 25., pero con la particularidad de que se puede chatear con personas nativas del idioma que se desea aprender, desconocidas totalmente. Es muy fácil de usar, con solo descargar la aplicación, ingresar su idioma natal e idioma que se desea aprender, su nivel actual del mismo y eso basta para ser capaz de ver una lista de los hablantes extranjeros nativos disponibles, se puede escoger entre sus más de 100 idiomas existentes. En otras palabras, Hello Talk es un motor de búsqueda avanzada de intercambio de idiomas, traductor instantáneo, además de que se puede crear una base de datos específica para: palabras, frases, sonidos, correcciones gramaticales, por mencionar algunas. Hello Talk cuenta con tecnología VoIP de última generación por lo que se pueden realizar y recibir videollamadas gratuitamente. Hello Talk actualmente se encuentra disponible para los sistemas operativos iOS y Android, gratuitamente (HelloTalk Inc., 2017).

Figura 25. Aplicación móvil *Hello Talk* para conversar en otros idiomas

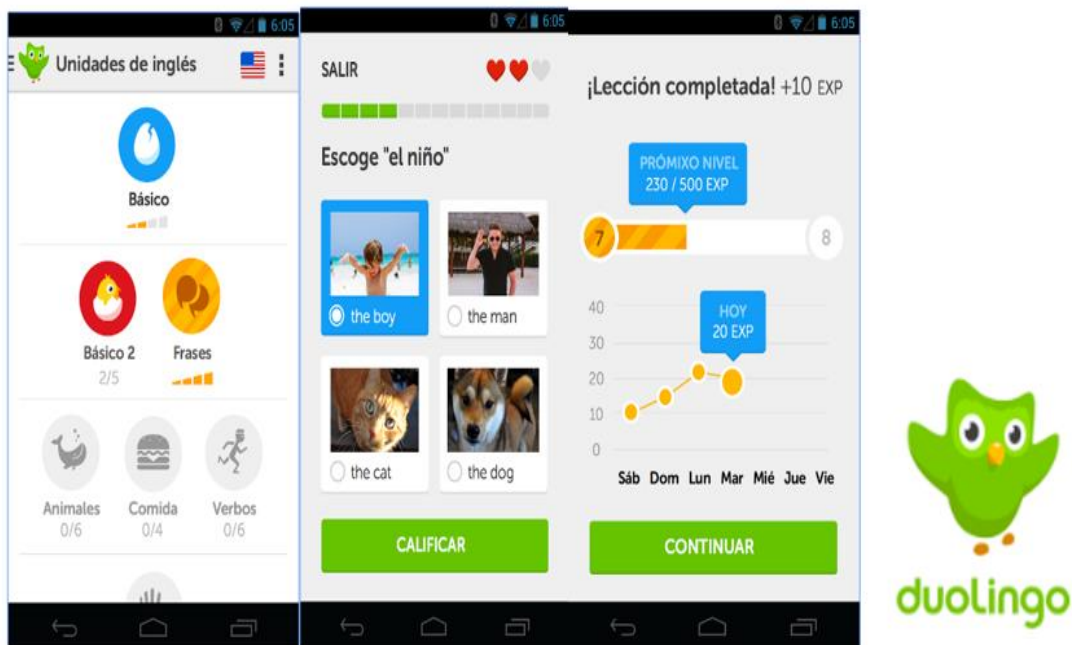


Fuente: *Hello Talk Inc., 2017.*

6. Duolingo

Duolingo es una App para aprender idiomas, como lo muestra la Figura 26., con la característica de mostrar algunas particularidades de un videojuego (se tienen vidas, se ganan monedas, se avanza de nivel), lo que lo hace entretenido y divertido, ya que es un método que se evalúa científicamente la eficacia del mismo, continuamente. Duolingo está disponible para los sistemas operativos iOS y Android, gratuitamente. Sin embargo, algunos productos integrados tienen un precio de \$19.00 a \$ 219.00 USD por elemento (Duolingo, 2017).

Figura 26. Aplicación móvil Duolingo para aprender idiomas



Fuente: Duolingo, 2017.

Se pueden destacar las siguientes Apps, siendo las más cercanas en algunos aspectos al concepto original de idea, que se desea desarrollar para el presente proyecto de tesis, en estas mismas particularidades, se podrían indicar como la competencia más cercana al prototipo de App *Draw-Hánzí*, que existen actualmente.

A. Skritter Chinese

Figura 27. Aplicación Móvil Skritter para aprender idioma chino y japonés.

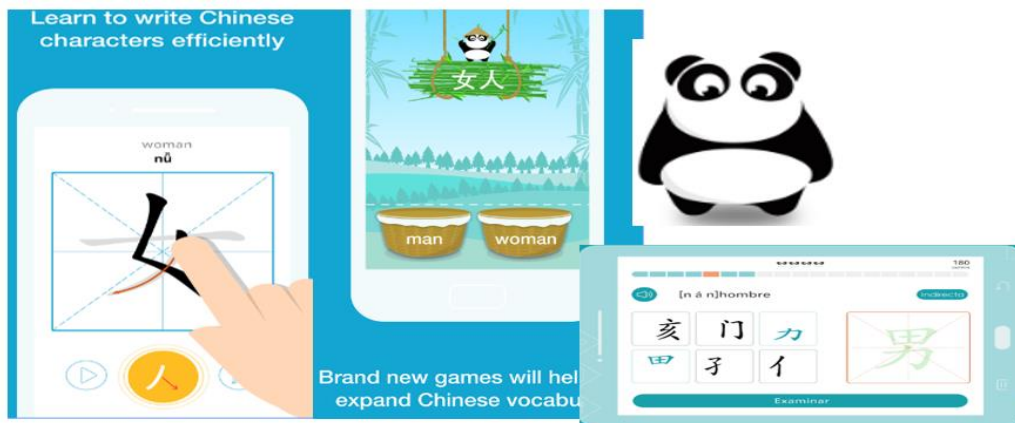


Fuente: Skritter, 2008-2017

Skritter es una App, para aprender a leer y escribir el idioma chino simplificado y tradicional, de entre una lista de 10,000 caracteres, 400,000 palabras y 17,000 archivos de audio nativos; práctica fuera de línea, sincronización automática cuando se conecte nuevamente a Internet, como se ilustra en la Figura 27. Puede ser instalado en sistemas operativos Android e iOS 8 o superior y es compatible con iPhone, iPad y iPod touch. Skritter Precio: \$14.99 USD / mes - \$ 99.99 USD / al año (Skritter, 2017; ChinesePod Limited, 2016).

B. ChineseSkill

Figura 28. Aplicación móvil para aprender el idioma chino Chinese Skill.



Fuente: ChineseSkill Co., Ltd., 2017.

Chinese Skill es una App, para aprender el idioma chino simplificado y tradicional, en pinyin, caracteres o ambos, que basa su plan de estudios en un juego estructurado y optimizado, para estudiantes con idioma natal inglés, la cual ofrece, escritura de caracteres chinos a mano diseñada, audios con pronunciación nativa en dos velocidades diferentes, además de enseñar gramática y sintaxis. Tiene lecciones de situaciones cotidianas, por ejemplo, ir de compras, amigos y vida social, saludo y presentaciones, ilustrada en la Figura 28. Sincroniza el avance en las lecciones a través de múltiples dispositivos de uso, además de que presenta la opción de uso sin internet. *Chinese Skill* está disponible para los sistemas operativos iOS 8 o posterior, es compatible con iPhone, iPad y iPod touch tiene un precio de \$9.99 USD/ al mes, \$19.99 USD / por tres meses, \$33.99 USD / por seis meses y \$59.99 USD /por un año, y en Android es, totalmente gratis. Sin embargo, algunos productos integrados tienen un precio de \$59.00 a \$ 1, 199.00 USD por elemento (*ChineseSkill Co., Ltd., 2017*).

C. *Aipoly Vision*

Figura 29. Aplicación móvil *Aipoly Vision*



Fuente: Aipoly Inc., 2017.

Aipoly Vision es una App, de reconocimiento de objetos y/o colores, para personas con discapacidades visuales ciegos y daltónicos (deficiencia visual o deficiencia al color), dentro de las funciones que maneja están ilustradas respectivamente en la

Figura 29.: el reconocimiento de objetos a la velocidad de 3 veces por segundo, reconocimiento de plantas y animales, lectura de texto, reconocimiento de alimentos, reconocimiento de colores, en diferentes idiomas como lo son: el inglés, español, francés, alemán, japonés y árabe. Tiene la función adicional de detección de oscuridad, lo que enciende en automático el flash de la cámara del dispositivo, además de poder accionar sus funciones vía voz y/o c vía un clic, además de poder usarse sin la conexión a internet necesariamente. *Aipoly Vision* está disponible para los sistemas operativos móviles iOS 10.0 o posterior, además de ser compatible con iPhone 5s, iPhone 6, iPhone 6 Plus, iPhone 6s, iPhone 6s Plus, iPhone SE, iPhone 7, iPhone 7 Plus, iPad Air, iPad Air Wi-Fi + Cellular, iPad mini 2, iPad mini 2 Wi-Fi + Cellular, iPad Air 2, iPad Air 2 Wi-Fi + Cellular, iPad mini 3, iPad mini 3 Wi-Fi + Cellular, iPad mini 4, iPad mini 4 Wi-Fi + Cellular, 12.9-inch iPad Pro, 12.9-inch iPad Pro Wi-Fi + Cellular, 9.7-inch iPad Pro, 9.7-inch iPad Pro Wi-Fi + Cellular, iPad (5th generation), iPad Wi-Fi + Cellular (5th generation), and iPod touch (6th generation), Snapdragon 800 o más y en Android 5.1 o superior tiene un precio de \$ 4.99 USD / al mes (Aipoly Inc. , 2017).

D. TapTapSee

Figura 30. Aplicación móvil *TapTapSee*



Fuente: CamFind Inc., 2016.

TapTapSee es una App, para personas con discapacidad visual, ayudándoles a identificar los objetos que se encuentran en su vida diaria, como se ilustra en la Figura

30. Solo se requiere de tomar una foto del objeto que se desea identificar y vía audio se obtendrá la identificación del mismo en diferentes idiomas como son: inglés, árabe, croata, checo, danés, japonés, noruego, polaco, rumano, ruso, chino simplificado y tradicional, español. TapTapSee está disponible para los sistemas operativos móviles iOS 7 o posterior, es compatible con iPhone, iPad y iPod touch y Android 4.0 o superiores, gratuitamente. Sin embargo, en este último existen productos en precios que van desde \$130.11 a \$ 1, 225.35 USD por elemento (CamFind Inc., 2016).

E. *Phonic Tricksters*

Figura 31. Aplicación móvil *Phonic Tricksters*



Fuente: *Pearson Education Limited, 2014.*

Phonic Tricksters es una App con AR, que asemeja un videojuego en donde los *Tricksters* (tramposos), se han escapado de una isla el saber y solo pueden ser atrapados cuando se les toma una fotografía y pueden escapar cuando se contesta una pregunta fonética, además de mostrar el audio de 20 letras, además de tener una interactividad con los niños, ya que los motiva a perseguir el conocimiento por toda su casa, como se muestra en la Figura 31. *Phonic Tricksters* está disponible para el sistema operativo móvil Android 4.0.3 y versiones superiores, gratuitamente. Sin embargo, algunos productos en compra tienen un precio de \$ 14.42 USD por elemento (*Pearson Education Limited, 2014*).

Capítulo III

CASO DE ESTUDIO: DRAW- HÁNZÍ

3. CASO DE ESTUDIO: DRAW-HÁNZÍ

3.1. Recolección de Información.

Con el fin de poder obtener los requerimientos iniciales para la producción del prototipo de App. Caso estudio: Draw-HánzÍ, fue necesario realizar una investigación documental referente a dos áreas de dominio:

3.1.1. Apps educativas y/o generación de Aprendizaje

Según la Organización de las Naciones Unidas para la Educación la Ciencia y la Cultura (*UNESCO*, por sus siglas en el idioma inglés), considera que el uso de las Tecnologías de la Información y la Comunicación (*TIC*, por sus siglas), ayuda a lograr el acceso universal a la educación, mejoran la igualdad y la calidad de las mismas; también contribuyen al desarrollo profesional de los docentes y mejoran la gestión, la gobernanza y la administración de la educación, siempre y cuando se apliquen las políticas, las tecnologías y las capacidades adecuadas (UNESCO, 2017).

El aprendizaje continuo, se define como aprendizaje sin solución de continuidad a lo largo de diferentes contextos, incluidos los entornos formales e informales. En la hipótesis ideal del aprendizaje continuo, el educando se sirve de manera oportunista de distintos tipos de tecnología, beneficiándose de las ventajas particulares de cada una de ellas (p.ej. la movilidad del teléfono inteligente) para mantener la continuidad de la experiencia de aprendizaje a través de dispositivos y entornos variados.

Históricamente ha habido una división, significativa entre el aprendizaje formal que se produce en un aula y el aprendizaje informal que acontece en el hogar o en ambientes comunitarios. Numerosos expertos están investigando cómo el aprendizaje móvil podría ayudar a eliminar esa barrera y cerrar la brecha que separa el aprendizaje formal del informal.

En Singapur, por ejemplo: se ha estudiado el empleo de las tecnologías móviles para facilitar el aprendizaje de los alumnos de primaria en diferentes contextos y ubicaciones. El proyecto Leveraging Mobile Technology for Sustainable Seamless Learning in Singapore Schools (SEAMLESS), fue un estudio longitudinal, efectuado a lo largo de tres años, pionero en el empleo de dispositivos móviles como ejes de aprendizaje para integrar los instrumentos del aprendizaje personal y ofrecer un único lugar en el que guardar el historial de aprendizaje y los recursos de cada alumno.

Los resultados de dicha investigación han sido marco de diseño, y de ayuda para la identificación de diez dimensiones del aprendizaje continuo, a saber: aprendizaje formal e informal, aprendizaje personalizado y social, aprendizaje a lo largo del tiempo, aprendizaje en distintos lugares, acceso al conocimiento desde cualquier lugar, mundo físico y mundo digital, tipos de dispositivo múltiple, tareas de aprendizaje múltiple, síntesis de conocimientos y modelos pedagógicos múltiples. Este tipo de enfoque holístico del aprendizaje, facilitado por los omnipresentes dispositivos móviles, debería ser un objetivo global para el futuro de la educación mundial (UNESCO, Op. Cit).

La Tecnología Educativa.

Las innovaciones recientes de las tecnologías móviles, se han centrado sobre todo en la creación de contenidos digitales, básicamente en forma de libros de texto digitales a los que se accede mediante lectores electrónicos, y el desarrollo de Apps o plataformas de software, para acceder a recursos educativos a través de dispositivos móviles (Shuler, Winters, & West, 2013).

Con base en la investigación del presente trabajo de tesis, se puede afirmar, que las TIC, específicamente, los dispositivos móviles y las Apps; han tenido un impacto revolucionario, firme y fuerte en el aprendizaje y/o educación, así como en el desarrollo permanente de competencias, desde un punto de vista “alumno / educando”, en cuanto a su motivación, atención, y su actitud receptiva, así como en el desarrollo de habilidades claves, como el sentido de la iniciativa, autodeterminación “la capacidad de aprender por aprender”, la competencia digital, además del trabajo en equipo presencial y/o virtual y la creatividad e innovación.

La integración de las TIC, en las aulas y en el día a día, ha favorecido a la personalización tan necesaria del aprendizaje, respetando los ritmos característicos

de cada individuo y facilitando de este modo su acertada participación, para con el mismo. Es en este sentido, el uso de herramientas TIC, cualesquiera que estas sean, vuelve al aprendizaje inclusivo, flexible y tan profundo como se desee adquirir, en la diversidad de personas con o sin necesidades educativas especiales, volviendo dichas herramientas, el mejor elemento promotor en el cambio educativo actual.

Cuando los alumnos y los docentes, utilizan cada vez más y para un sinnúmero de actividades de diversas asignaturas, a los dispositivos móviles con fines educativos, trabajando e interactuando con Apps, generadoras de aprendizaje, por ejemplo: la búsqueda de información en Internet, solo por el hecho de estar lo más informado posible: crea una cultura de colaboración y acción de aprendizaje (alfabetización) digital vital, ya sea nivel personal y/o profesional.

3.1.1.1. Aprendizaje INMERSIVO

¿En qué consiste el método de aprendizaje inmersivo? Consiste en una técnica, que se centra en sumergirse de manera concentrada en una actividad y/o tarea, y que permanece en el ejercicio de esta misma ya que se le dificulta la salida, por una situación ajena a la actividad.

En el contexto de las tecnologías innovadoras, el aprendizaje inmersivo es muy factible y viable, gracias a que cambia la perspectiva del usuario y acelera el aprendizaje para el mismo. Se dice, que el aprendizaje inmersivo tiene su origen y/o función del siguiente pensamiento:

*“Me lo contaron y lo olvidé; **lo vi y lo entendí; lo hice y lo aprendí**”- Confucio.*

Ahora bien, el método de inmersión lingüística, trata de un curso intensivo de aprendizaje de idiomas, donde los estudiantes, que desean aprender una lengua extranjera, conviven presencialmente las 24 horas del día, con nativos del idioma, realizando actividades cotidianas, orientadas a mejorar la capacidad comunicativa de los estudiantes en una lengua extranjera.

Dado que, este método proporciona una serie de beneficios, distintos a los que se adquieren en la enseñanza tradicional académica de las lenguas extranjeras actualmente como los son:

- Mejora la confianza al hablar un idioma extranjero.
- Mejora las habilidades comunicativas orales (escuchando “listening” y hablando “speaking”).
- Aumenta la motivación en los alumnos de seguir aprendiendo.
- Se supera el miedo a interactuar y hablar un idioma extranjero.

A su vez, el método de aprendizaje inmersivo lingüístico mediante el uso de realidad virtual, la AR o bien el uso de videos y audio en 360°, cuyo principio es la asimilación de conceptos; está basada en la metodología del **Learning By Doing** (*aprender haciendo*), de **quien Roger Schank** es el autor y cuyo principio establece: que los alumnos deben adquirir competencias en el camino a alcanzar una meta u objetivo. Para él, el problema de los métodos de aprendizaje actuales, es que los estudiantes no están desarrollando competencias, ya que la enseñanza está enfocada únicamente en compartir el conocimiento. Según algunos estudios, este método de aprendizaje tiene un porcentaje de retención del conocimiento del 90%, mientras que el resto de los formatos tradicionales es de entre 10% y 20%. Un ejemplo, (algo extremo pero ilustrativo), utilizado por Schank, es el de la biología; él dice que a los alumnos se les enseña, que los animales tienen un hígado y un corazón, pero a menos que se les enseñe, qué hacer con este conocimiento, los alumnos pronto olvidarán lo aprendido (Legorreta, 2015).

3.1.2. El idioma CHINO

Actualmente vivimos en un mundo globalizado, donde la tecnología y el acceso a la información, se han extendido vertiginosamente, sin exclusión de casi ninguna parte del mundo. Con relación a esto, y por la trascendencia de la misma en un sin número de áreas, los expertos en diferentes campos de acción, siempre y en todo momento, recomiendan aprender como mínimo un segundo idioma extranjero cualquiera que esta sea, de forma autodidacta o instruida, sacando el mejor provecho del conocimiento que se comparte en Internet y que está disponible en algunos caso gratuitamente o por un acordado bajo costo, para cualquier persona que quiera y/o pueda adoptarla, como una oportunidad de sobresalir en cualquier aspecto de su vida.

Sin embargo, con las particularidades importantes que implica el aprendizaje de un segundo idioma, ya que muchas veces se requiere de técnicas específicas, denominadas de inmersión, memorización y/o experiencia, desarrolladas para que cada una de las personas, que aprendieron una lengua natal de manera natural y por intuición, desarrollen las estrategias y estilos necesarios que les permitan adquirir el conocimiento de una manera útil, sencilla, práctica y asequible de un segundo idioma o más si así se desea.

Es decir, que existe en cada método de aprendizaje factores clave como lo son: la edad, la inteligencia, las capacidades motoras, la motivación, por mencionar algunas. Enfatizando también que mientras más joven, se tenga la oportunidad y asequibilidad, de aprender un segundo idioma nuevo y/o paralelo, a su lengua natal será mil veces mejor, debido a que el cerebro es más permeable al conocimiento, absorbiéndolo fácilmente, cuestión que se complica demasiado, al paso del tiempo.

Entonces, la pregunta más obvia es: ¿Cuáles son los idiomas más útiles?, actualmente, ¿Qué idioma le permite interactuar de manera más integral con la humanidad, a nivel global?

Para definir la eficacia de un idioma (por extensión, de una cultura), debemos considerar las puertas “oportunidades”, que abre. A grandes rasgos, un idioma eficiente brinda cinco oportunidades:

1. Geográfica: la capacidad de viajar libremente.
2. Económica: la capacidad de participar en una economía.
3. Comunicación: la capacidad de entablar un diálogo.
4. Conocimiento y medios: la capacidad de consumir conocimiento y contenido de los medios.
5. Diplomacia: la capacidad de abordar relaciones internacionales.

De acuerdo con, las oportunidades anteriores, se puede construir un índice para comparar o clasificar los idiomas, de acuerdo a su eficiencia en distintos dominios. El índice de utilidad de idiomas (*Power Language Index; PLI*, por sus siglas en inglés), utiliza 20 indicadores para medir cuál influyente es un idioma, como lo muestra la Tabla 13., (Chan, 2016).

Tabla 13. Clasificación del índice de potencia de un lenguaje.

RANK	SCORE	LANGUAGE	NATIVE (MM)	GEOGRAPHY	ECONOMY	COMMUNICATION	KNOWLEDGE & MEDIA	DIPLOMACY
1	0.889	English	446.0	1	1	1	1	1
2	0.411	Mandarin*	960.0	6	2	2	3	6
3	0.337	French	80.0	2	6	5	5	1
4	0.329	Spanish	470.0	3	5	3	7	3
5	0.273	Arabic	295.0	4	9	6	18	4
6	0.244	Russian	150.0	5	12	10	9	5
7	0.191	German	92.5	8	3	7	4	8
8	0.133	Japanese	125.0	27	4	22	6	7
9	0.119	Portuguese	215.0	7	19	13	12	9
10	0.117	Hindi*	310.0	13	16	8	2	10

Nota 1: * Si se consideran todos los dialectos / idiomas chinos (mandarín, que es el más grande), como uno solo, no cambiaría su orden en la lista. Sin embargo, si el Urdú y el hindi (todos los dialectos hindi), se toman como uno solo, estaría más arriba que el portugués y el japonés.

Fuente: Chan, Op. Cit.,2016.

Existe una enorme posibilidad, de que las personas, alrededor de usted y/o más cercana a nosotros (familia, hijos amigos, conocidos, colegas, por mencionar algunos), se relacionen cada día más, con otros dialectos e idiomas, si se comparte con ellos, el interés asequible por otras culturas, generando de esta forma, que los cercanos más directos, se interesen por el idioma como una herramienta de comunicación y comprensión del mundo en general.

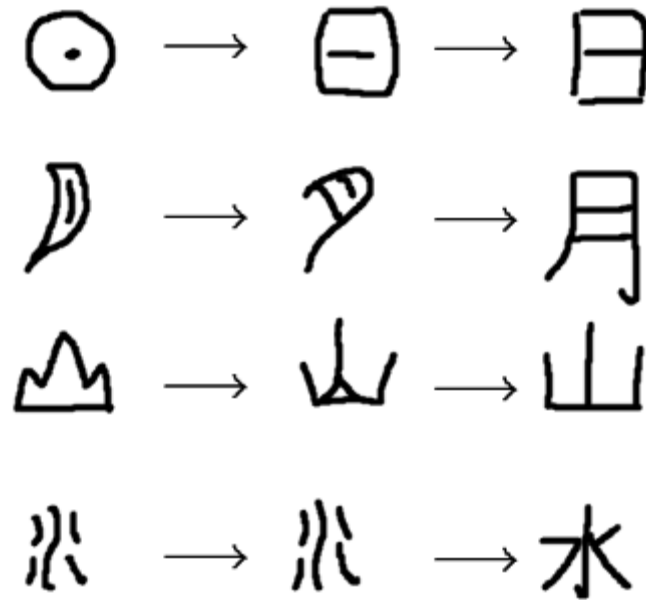
Póngase por caso, el idioma chino (hanyu 汉语, denominado en el idioma chino mandarín) y su escritura, se originó hace más de tres mil años aproximadamente, como representación pictórica de ideas simples. Para muchos expertos multidisciplinarios es la lengua más hablada del mundo actualmente, con su más de 1,100 millones de hablantes alrededor del mundo⁷, aproximadamente (una quinta parte de la población mundial), como su lengua natal o materna. El sistema de escritura chino, se compone de miles de signos gráficos a los que en español conocemos como *caracteres*. Sin embargo, el término chino para dichos caracteres es 汉字 *hànzì*.

Los caracteres chinos, más primitivos eran meras representaciones pictóricas de ideas simples, por ejemplo, los caracteres que representaban conceptos tan básicos como el sol o la luna consistían en dibujos que intentaban evocar la imagen de estas

⁷ El número de hablantes fue obtenido del artículo “¿Cuáles son los idiomas más hablados en el mundo?”, por Octavio Noguez, 12-02-2017 en <https://www.merca20.com/cuales-los-idiomas-hablados-en-mundo/>.

realidades concretas de la naturaleza. Muchos de estos caracteres de origen pictórico continúan en uso en el idioma actual, aunque su estética gráfica ha evolucionado en múltiples ocasiones, tal como se muestra en la Figura 32., de modo que, en su forma actual, apenas son ya reconocibles como dibujos representativos.

Figura 32. Evolución histórica de los caracteres chinos.



Nota 2: En esta imagen se ve la evolución histórica de cuatro caracteres sencillos, con los significados de *sol*, *luna*, *montañas* y *agua* desde las formas más arcaicas de la columna izquierda hasta las formas actuales de la columna derecha.

Autor: Ángel José Riesgo, 2011.

Algunos expertos, han expresado y denominado al sistema de escritura del idioma chino cómo: el más fascinante, hermoso, lógico y científico del mundo. ¿Por qué?, hacen tal aseveración, es debido a que los caracteres chinos reúnen sonidos, significados e imágenes visuales en un conjunto “coherente”, por lo tanto, cada trazo posee su propio significado.

Todos los caracteres chinos tienen sus orígenes y de acuerdo con la etimología tradicional, se pueden clasificar en seis métodos de escritura (composición) denominada **Liu Shu**, siendo los siguientes:

- Pictogramas sencillos (xiang xing): Representaciones gráficamente estilizadas de objetos concretos.
- Ideogramas sencillos (zhi shi): Expresiones simbólicas de conceptos abstractos, fenómenos y representaciones.
- Ideogramas compuestos y/o complejos (hui yi): Representaciones gráfico-simbólicas que combinan dos o más caracteres para formar uno nuevo, en donde cada carácter aporta un sema o unidad de sentido diferente, por lo que podrían denominarse como una composición de significados.
- Compuestos fonéticos y semánticos “Formas y sonidos” (xing Sheng): la combinación de un elemento con significado visual o grafía y otro de carácter fonético. La mayor parte de los caracteres chinos son de este tipo, el elemento que denota su significado se llama *radical*, que normalmente va a la izquierda y el que indica la pronunciación se llama *fonema*.
- Transición y/o extensión etimológica (zhuan zhu): consiste en emplear caracteres con un significado derivado etimológicamente del sentido original, en este tipo se pueden incluir palabras extranjeras.
- Supuesto y/o préstamo falso (jia jie): comprende caracteres que se toman prestados para representar una palabra con la que sólo guardan relación por su similitud fonética (Galan Sempere, 2013).

Las propiedades que se refieren al trazo de los caracteres, tanto para escribirlos como para encontrarlos en los diccionarios de una forma práctica y sencilla, se considera, otro método, denominado radicales. Sin embargo, este método tiene cierto grado de complejidad, ya que tanto para escribirlos como para buscarlos en los libros especializados, hay que conocer perfectamente, el número de trazos de cada carácter, su figura y el orden sucesivo dentro de la escritura.

Debido a que el sistema de escritura da la sensación, de ser extremadamente complicado, el hecho de simplificar los trazos, lo hace toda vía más confusa, ya que, si se elimina o se cambia algún trazo, no solo se oscurece su origen (según las tradiciones y costumbre de la cultura oriental), sino también su categoría de carácter, a la vez también se le despoja de gran parte de sus características.

La Figura 33., muestra como en la actualidad para usos prácticos tanto de enseñanza, como de comunicación, por mencionar algunos, se puede dividir a la escritura de la caligrafía china en tres formas: el completo (o tradicional), el simplificado y el pinyin (romanizado).

Figura 33. Comparativo de los tipos de caligrafía china

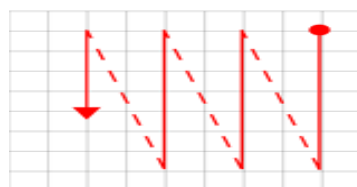


Autor: Leon Z. Lee, 1999.

Donde:

- La caligrafía china TRADICIONAL: Es útil para comprender mejor la literatura antigua y también puede evitar la confusión (muy común) entre caracteres. Se debe tomar en cuenta que sirve especialmente para el mundo de los negocios y/o para comunicarse por escrito con más personas. Básicamente es la forma más intrincada y con muchos más trazos, cuya forma es generalmente cuadrada, se encuentran en un espacio tipográfico de forma cuadrada, la cual originalmente se escribía de forma vertical y se leía de arriba hacia abajo y de derecha a izquierda como lo muestra la Figura 34. Los lugares donde actualmente se usan los caracteres tradicionales son: *Taiwán, Hong Kong y Macao.*

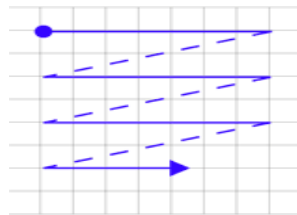
Figura 34. Ejemplo del sentido de escritura tradicional.



Autor: LinotYpE.com, 2017.

- La caligrafía china SIMPLIFICADA: Son caracteres mucho más sencillos de aprender, memorizar, leer y de escribir, se usan y de forma más reducidas, con menos número de trazos. Tienen una forma casi cuadrada, en un espacio tipográfico de forma cuadrada. Además de que ayudan a incrementar la tasa de alfabetización y se ha consagrado como el un idioma más estándar, se escribe de izquierda a derecha y de arriba hacia abajo como se muestra en la Figura 35. Los lugares donde actualmente se usan los caracteres tradicionales son: *China continental (RPC), Malasia y Singapur.*

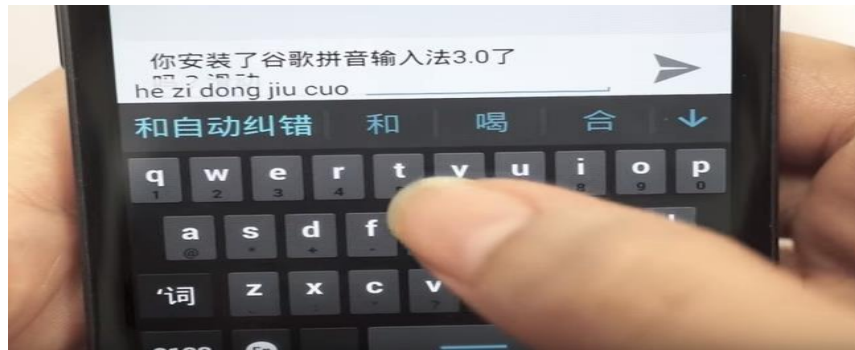
Figura 35. Ejemplo del sentido de escritura simplificada



Autor: LinotYpE.com, 2017.

- La caligrafía china PINYIN ROMANIZADA: Es un sistema de transcripción fonética del chino mandarín, como para usar tanto a la hora de escribir por computadora o en cualquier dispositivo móvil (en lugar de tener un teclado lleno de caracteres chinos, se introduce el pinyin y a medida que se va escribiendo, aparecen los diferentes caracteres que corresponden a la transcripción fonética como lo muestra la Figura 36). Si se observa, un diccionario o incluso el Google Translate, se puede observar como una determinada palabra, además de su traducción a carácter, tiene una transcripción fonética pinyin como lo muestran la Figura 37 y 38 respectivamente, lo que permitirá que una vez que se conozca la forma en que se lee la transcripción fonética, asegurar la facilidad de poder pronunciar la palabra sin problemas, ya que se les puede observar en diferentes lugares como son: los mapas del metro, señales de tráfico o manuales de enseñanza para niños hoy en día, esta forma de escritura nunca ha tenido un estatus de escritura oficial, ya que en china se considera que los caracteres chinos son un patrimonio cultural insustituible.

Figura 36. Ejemplo de escritura Pinyin en teclado de dispositivo



Autor: Google Inc., 2017.

Figura 37. Ejemplo de traducción de una palabra en español a chino simplificado en Google Translate.



Autor: Google Translate, 2017.

Figura 38. Ejemplo de traducción de una palabra en español a chino tradicional en Google Translate.



Autor: Google Translate, 2017.

El pinyin tiene su grado de complicación, debido a la importancia de los cuatro tonos existentes, aprenderlos es de suma importancia, ya que, de no utilizarlos correctamente, hará que los nativos chinos no entiendan o se confundan de significado. El tono indica cómo se tiene que modular la voz cuando se pronuncian las vocales de las sílabas, estas sencillas variaciones de tonalidad de la voz, a la hora de pronunciar las sílabas chinas pueden determinar que se diga una palabra en lugar de otra y generar muchos malentendidos (considerados de muy mal gusto y educación en los países orientales).

La existencia esencial de los cuatro tonos en chino como lo muestran la Figuras 39 y 40., respectivamente, se debe principalmente a qué, el idioma chino, tiene un número limitado de sílabas, aproximadamente unas 400, muy pocas en comparación con las 12,000 que existen en el idioma inglés. Es por esta razón, que hay muchas palabras homófonas (palabras que se pronuncian igual, pero que tienen diferente significado), aquí es donde el sistema de tonos, multiplica la variedad fonética del idioma chino.

Figura 39. Ejemplo de los cuatro tonos de chino mandarín

ā á ǎ à

Autor: Linguese, 2015.

Donde:

- Primer tono: No tiene variaciones en el tono de voz, este debe ser constante y alargado.
- Segundo tono: La tonalidad es ascendente, se comienza con una tonalidad suave y se va subiendo, se puede ejemplificar en español, a la tonalidad utilizada cuando se hace una pregunta, por ejemplo, ¿Por qué?
- Tercer tono: la tonalidad de voz es alta al inicio y va descendiendo por un instante para subir enseguida.
- Cuarto tono: El tono de voz es seca y fuerte, de arriba hacia abajo.

Figura 40. Ejemplo de significado, dependiendo el tono del carácter.



Autor: Club de Lectores Instituto Confucio, 2017.

Por esto, los estudiosos chinos, han atribuido siempre una gran importancia a la caligrafía, por múltiples motivos, así como debido a sus diversos beneficios físicos y espirituales, para quienes la practican a manera de infundir paciencia, concentración, disciplina y perseverancia. Su ejercicio puede llegar incluso a refinar la personalidad y cambiar la forma de ver la vida (de acuerdo a los usos y costumbres de ese país oriental). Por ejemplo, en Taiwán, es una de las asignaturas, con mayor importancia de impartición. Ahora bien, los calígrafos chinos siempre han utilizado brochas, tinta, papel y tintero de piedra, para crear verdaderas obras de arte, con la caligrafía. A los antes mencionados instrumentos se les conoce como “los cuatro tesoros de estudio” (文房四宝; Wénfáng Sibǎo), ilustrados en la Figura 41., respectivamente.

Figura 41. Ejemplo de instrumentos para la escritura caligráfica china.



Autor: Exposición “El libro en China” de la biblioteca de la facultad de filología edificio A, de la Universidad Complutense, 2017.

La forma, el tamaño, el tipo de cabello de la brocha, el color (intensidad) y la densidad de la tinta, así como su nivel de absorción y la textura superficial del papel, son los factores físicos principales que afectan, el aceptable resultado final considerablemente, como se muestra en la Figura 42. En la caligrafía china, todo cuenta, pero existe un compendio de cinco criterios que todo buen calígrafo debe seguir para que su trabajo pueda llegar a ser considerado una obra de arte verdadera o ya como mínimo con la calidad y respeto que esta cultura necesita. Estos son los siguientes criterios: el carácter debe estar escrito correctamente, debe ser legible y conciso, los caracteres deben estar en un contexto apropiado y finalmente, deben estar estéticamente creados.

Figura 42. Ejemplo de escritura caligráfica china.



Autor: Actualizta ´t, Framepool, 2008.

- La correcta escritura de los caracteres.

El papel (patrón) para aprender a escribir o para ejercer la caligrafía china, recibe el nombre de “tián zì gé (田字格)” y hoy en día, se sigue empleando en muchas escuelas de oriente, ilustrado en la Figura 43. En china la caligrafía se realiza sobre una cuadrícula denominada “mǐzìgé (米字格), que facilita el trazo de la escritura con la distribución equilibradamente correcta, que requiere cada carácter tradicional y/o simplificado (en occidente se utilizan de un tamaño más grande, del que realmente se utiliza en china), aunque cuando ya se adquiere la destreza y equilibrio necesarios, se pueden realizar los trazos correctamente en el interior de un espacio cuadrado imaginario, a modo de cuadrícula, como lo muestra la Figura 44.

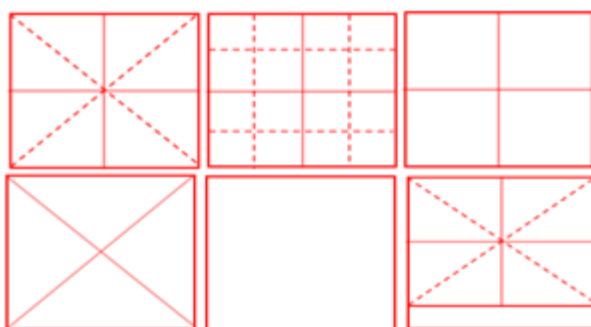
Figura 43. Ejemplo de caligrafía china sobre "Mǐzìgè"



Nota 3: El carácter " Nǐ " (en, Pinyin), se constituye de 7 trazos, es un pronombre cuyo significado puede ser: Tú, Tu, y/o Usted.

Autor: Chino-China.com, 2008.

Figura 44. Ejemplo de diferentes "Mǐzìgè's", utilizados para la enseñanza de la caligrafía china



Autor: Elaboración propia, 2017.

Aunque es recomendable practicar el trazado dentro de un Mǐzìgè al inicio de la enseñanza de la caligrafía china, también es válido que de considerarlo necesario practicarlos a mano con papel y bolígrafo o lápiz. En la práctica autodidacta sirve cualquier papel en blanco y/o se pueden descargar las múltiples plantillas de cuadrícula existentes en Internet. No obstante, si no se tiene a la mano papel y lápiz, también es válido que se emplee simplemente el dedo y "escribir / trazar" sobre una superficie lisa, inclusive sirve la palma de la otra mano opuesta, ya que tiene una forma más o menos cuadrada y en ella se pueden escribir los caracteres "sin tinta" alguna.

Con base a la experiencia autodidacta, se recomienda fundamental que se realice el ejercicio de escribir/ trazar un carácter, algunas veces (a la vez que se concentra en la escritura, componentes y significado del mismo) para asegurar la retención en la memoria del carácter que se está estudiando. La combinación de la imaginación y los movimientos físicos de la mano al escribir / trazar, como en cualquier método de aprendizaje, favorece notablemente el ejercitamiento y ayudan a fijarlo con mayor facilidad en la memoria.

- Los ocho principios de Yong

Desde el punto de vista de la forma (dibujo), los caracteres chinos parecen meros dibujos sin sentido y en algunas ocasiones dan la impresión de no ceñirse a ninguna regla de escritura, como las que se emplean en occidente o en el alfabeto romano. No obstante, aunque difieren mucho de las reglas gramáticas de otros lenguajes, los caracteres chinos (caligrafía), poseen sus reglas particulares, al trazarlos, por ejemplo, su debida distribución y un determinado orden, con la finalidad de que parezcan legibles (bellos y bien ordenados). Es en este punto donde se tornan muy complicados, aun para las personas que tienen la lengua nativa china.

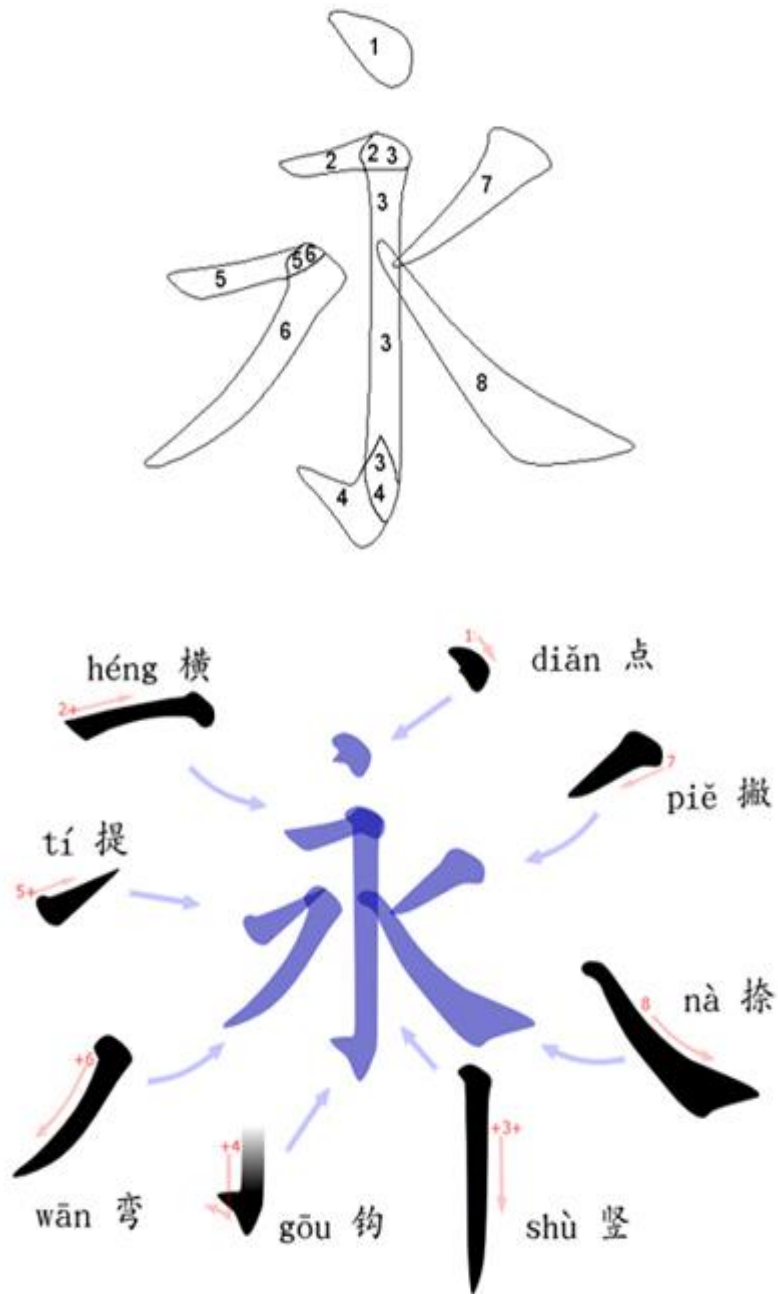
Al escribir / trazar un carácter chino hay que tener en cuenta lo siguiente:

- El número exacto de trazos de cada uno de los caracteres.
- La sucesión correcta de los trazos de cada uno de los caracteres.
- El tamaño correcto de cada trazo en cada uno de los caracteres.
- La posición o colocación correcta de los trazos de cada uno de los caracteres.
- La distancia correcta de un trazo en relación a otro trazo en cada uno de los caracteres.

Existen muchas teorías en la caligrafía china, pero una de las más importantes es la Eiji Happô “永字八法” (derivado de los Kakijun 書き順, en los trazos Kanji japoneses) o “las ocho leyes del carácter”, fue desarrollada por el monje budista Zhi Young (智永), durante el periodo de la Dinastía Sui “隋朝”, del 581 al 618 a.C. Por esta razón, dicha teoría, es conocida también como “los ocho principios de Yong 永”, siendo este carácter Yǒng 永, uno de los más importantes caracteres, tanto en el idioma chino como en el japonés, no es sólo un sinónimo de resistencia y perseverancia, que generalmente es traducido como “Eternidad y/o Permanencia”, sino que es uno de los caracteres que incluye, los ocho trazos fundamentales de la caligrafía como lo muestra la Figura 45., llamados ocho reglas o Happou 八法, de un total de entre 37 y 57 tipos de trazos de pincel. Sin embargo, 29 de ellos se derivan de esta regla de ocho y se creía que practicar estos ocho trazos frecuentemente, como lo hacía un calígrafo aseguraba la belleza de la escritura. Siendo los ocho trazos Yǒng, los siguientes:

1. Dian (punto 丶): De piedra o de proximidad (debido a su forma y/o apariencia de roca redondeada). Hay muchas maneras de escribir los puntos y aunque no lo parezca, esta es una pincelada exigente, es de hecho extremadamente difícil de escribir.
2. Héng (Horizontal 一): Movimiento a la derecha, también conocido como “un cabestro o mesa de jade”, de superficie lisa y uniforme. Es un trazo horizontal ligeramente ascendente, con un principio estricto y extremo, donde se compara el movimiento del pincel a la fuerza abrupta de las riendas de un jinete, para detener al caballo.
3. Shú (Vertical o Erecto 丨): Movimiento hacia abajo, también conocido como “ejercer o hacer un esfuerzo”. El trazo termina con una parada inmediata, imitando el mecanismo de bloqueo ballesta. El lado izquierdo de la línea vertical en el carácter debe ser hacia adentro y suavemente curvado, mientras que el lado derecho permanece recto. La hábil manipulación de la presión del pincel sobre el papel es la clave en este trazo y le da un aspecto sólido y rígido.
4. Gou (Gancho 丿 , 乙): Sugiere una subida o un gancho, por su forma hacia arriba. Va seguido de la línea vertical hacia abajo y de repente, gira hacia la izquierda y hacia arriba, perdiendo el grosor en el trazo gradualmente, pero sin perder vigor.
5. Tí (Ascendente o Elevado): Es un movimiento muy rápido como un látigo. Es un trazo horizontal, inclinado hacia arriba y con un adelgazamiento gradual hacia el final.
6. Zhé o Wan (Doblados o Curvos ㇇ , ㇈): Este trazo puede ser descrito como un barrido de curva, afilándose gradualmente hacia el final, llegando a parecer en su forma a la de un cuerno de rinoceronte.
7. Pie (Pendiente o con Caída a la izquierda ㇉): también conocido como el picoteo de un pájaro, es mucho más delgada y más delicada que la anterior “Wan o cuerno de rinoceronte”, una línea ligeramente curva hacia la izquierda y de manera descendente.
8. Ná (Presionar o Caída a la derecha ㇊): Es un movimiento hacia la derecha y bajando, emulando la forma de la espada Dao, con su característica hoja triangular. Este trazo es muy difícil de ejecutar al mismo tiempo que se hace hincapié en la energía y la expresividad del carácter.

Figura 45. Caracter Yǒng o Eiji Happô

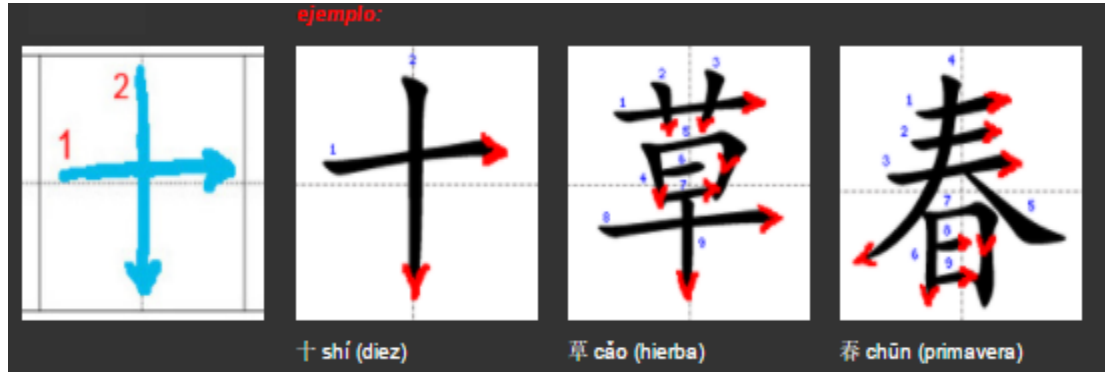


Autor: Smashing Magazine, 2010.

Una vez que se conocen los principales trazos de los caracteres chinos, se tiene que seguir los principios fundamentales (de vital importancia los primeros cuatro), así como sus tres excepciones, que marcan la secuencia correcta de los mismos, dentro del propio carácter, siendo las siguientes:

1. Primero el trazo horizontal y después el trazo vertical, ilustrado en la Figura 46.

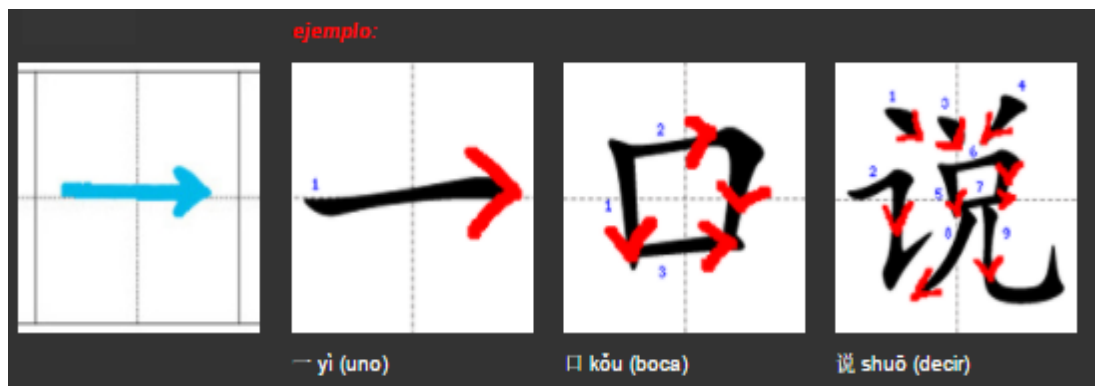
Figura 46. Ejemplo de trazo del primer principio fundamental de secuencia en un carácter chino



Autor: The Stroke Order Dictionary for Chinese characters, 2009.

2. Primero los trazos del lado izquierdo y luego los trazos del lado derecho, ilustrado en la Figura 47.

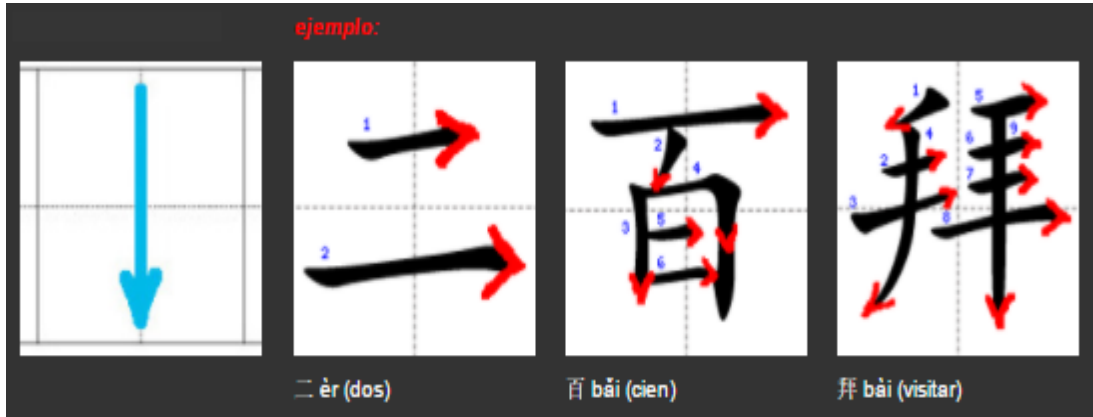
Figura 47. Ejemplo de trazo del segundo principio fundamental de secuencia en un carácter chino



Autor: The Stroke Order Dictionary for Chinese characters, 2009.

3. **Primero el trazo de superior y luego el trazo inferior, ilustrado en la Figura 48.**

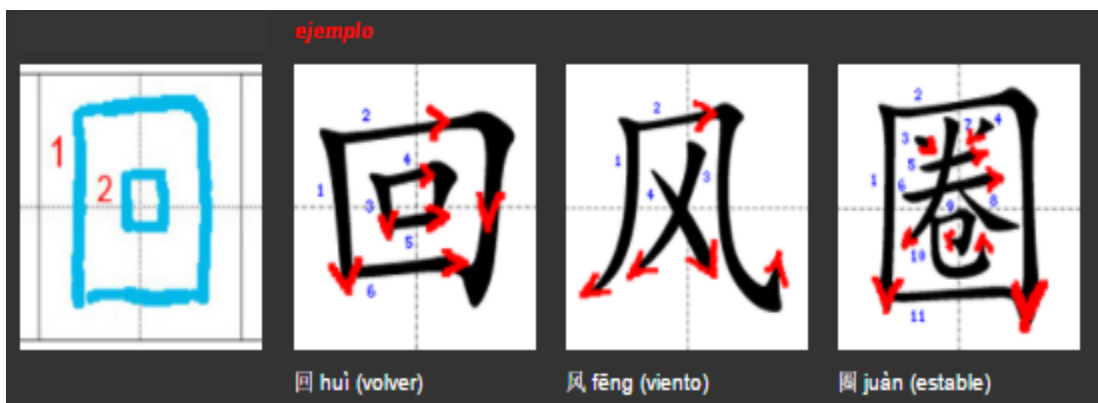
Figura 48. Ejemplo de trazo del tercer principio fundamental de secuencia en un carácter chino



Autor: The Stroke Order Dictionary for Chinese characters, 2009.

4. **Primero los trazos externos y luego los trazos internos, ilustrados en la Figura 49., (en los caracteres que hay en su estructura un cerco, se debe hacer por el lado izquierdo y el superior o entres sentidos: izquierda, inferior y derecha, deben escribirse según el principio: primero los trazos exteriores y luego los interiores).**

Figura 49. Ejemplo de trazo del cuarto principio fundamental de secuencia en un carácter chino



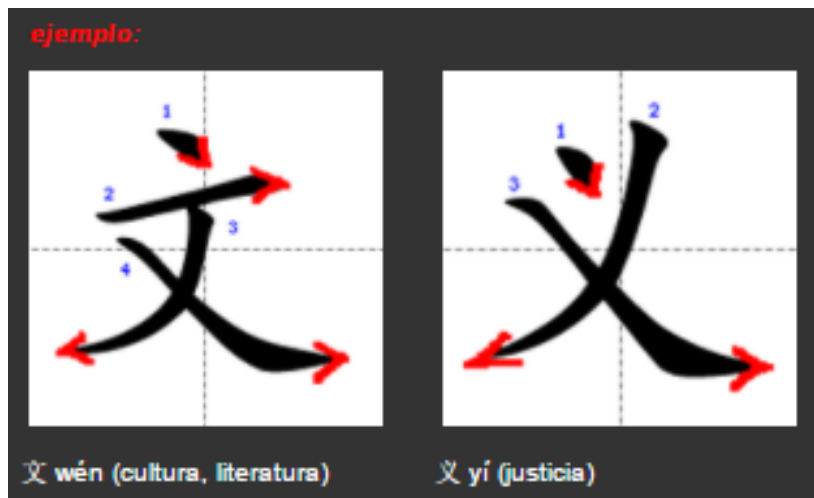
Autor: The Stroke Order Dictionary for Chinese characters, 2009.

5. **Primero la cola** (de arriba hacia abajo e izquierda) **y luego la escoba** (de arriba hacia abajo y derecha).
6. **Primero los trazos internos y luego los externos** (en los caracteres que hay en su estructura un cerco, se debe hacer por el lado izquierdo y el superior o entres sentidos: izquierda, inferior y derecha, deben escribirse según el principio: primero los trazos internos y luego los externos).
7. **Primero el trazo central y luego los trazos laterales.**
8. **El cierre al final.**

Excepciones:

1. **En caso de diagonales simétricas, de derecha a izquierda, serán trazadas primero las de la izquierda y luego las de la derecha, como lo muestra la Figura 50.**

Figura 50. Ejemplo de trazo de la primera excepción de los principios fundamentales de secuencia en un carácter chino



Autor: The Stroke Order Dictionary for Chinese characters, 2009.

2. **Frecuentemente los trazos largos de cerradura son las ultimas, como lo muestra la Figura 51.**

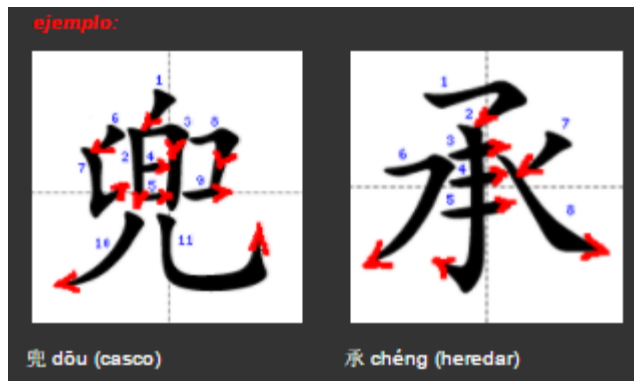
Figura 51. Ejemplo de trazo de la segunda excepción de los principios fundamentales de secuencia en un carácter chino



Autor: The Stroke Order Dictionary for Chinese characters, 2009.

3. Cuando el carácter en la parte de adentro (centro), lleva un cerco, se tienen que hacer primero los trazos de externos verticalmente simétricos, como lo muestra la Figura 52.

Figura 52. Ejemplo de trazo de la tercera excepción de los principios fundamentales de secuencia en un carácter chino



Autor: The Stroke Order Dictionary for Chinese characters, 2009.

- Caracteres compuestos.

No obstante, el sistema de escritura china, no tiene un alfabeto como tal, puesto que los caracteres chinos, no están formados por letras, únicamente, sino más bien por trazos. Cada carácter chino corresponde a una solo o más silabas, la mayoría de los caracteres chinos están en muchas ocasiones formados por dos o más grupos de

componentes, generalmente un radical semántico (de significado más o menos común) y uno fonético. Cada carácter chino es indivisible e invariable, sin embargo, por razones prácticas y de aprendizaje se pueden clasificar en simples y compuestos (según los trazos), ideográficos (sentido deducible de las partes) y un fonético (si contienen alguna indicación sobre la pronunciación).

Es decir, los caracteres chinos son unos de los lenguajes más antiguos, fueron compuestos en un inicio hace miles de años, y posteriormente fueron enseñados de generación a generación sucesivamente (el significado original, se perdió a través del tiempo), sin embargo, se siguen utilizando actualmente, por ejemplo, la palabra “DIOS”, en la antigüedad, los chinos ancestrales creían en un solo Dios, un ser supremo llamado “ShangTi”. La palabra Dios 上帝 (Shàngdì, por su pronunciación en Pinyin), está conformado por dos caracteres o silabas, que se escriben como una sola combinación de dos significados: “arriba” y “emperador”, como lo muestra la Figura 53.

Figura 53. Ejemplo de carácter chino de la palabra Dios



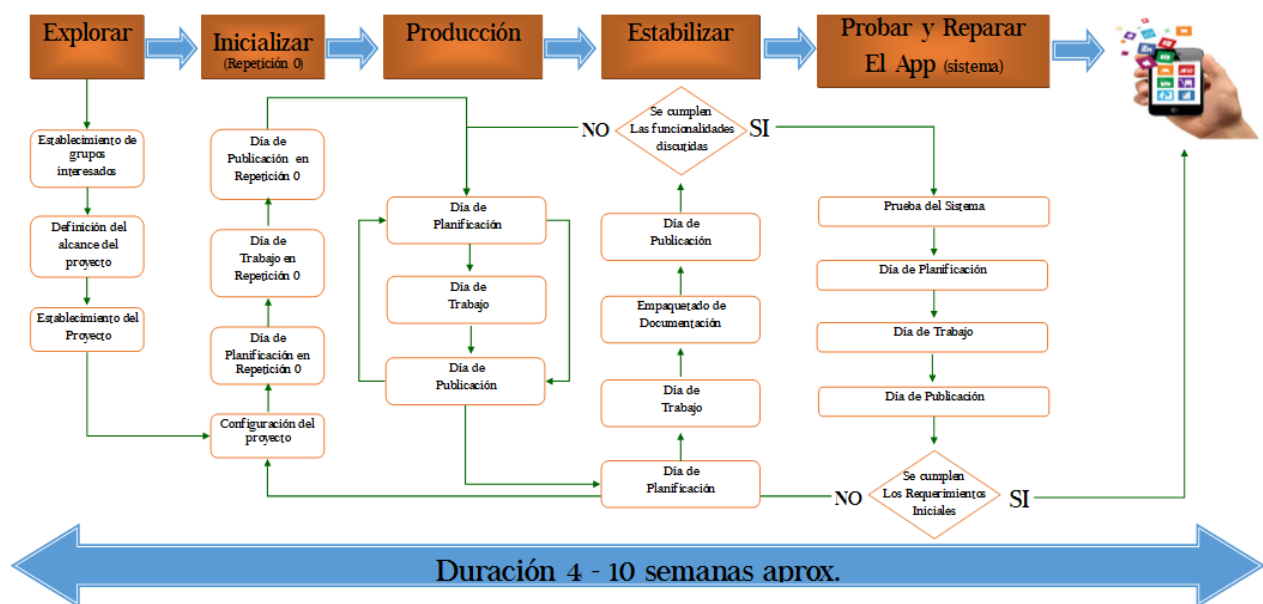
Autor: The discovery of Genesis: How the truths of genesis were found hidden in the Chinese language, 2012.

Por consiguiente, en términos generales, la historia de la caligrafía china puede dividirse en dos importantes periodos, el primero de ellos puede ser llamado como un periodo de evolución, así el segundo y último de ellos se puede denominar como el periodo de cambio en el estilo. El periodo evolutivo del estilo de caligrafía china, se desarrolla hacia la expresión de un estilo más artístico, que se interrelaciona con el estilo de la caligrafía. En el periodo de cambio de estilo de la caligrafía china, se centra en el significado, la estabilidad elegante y la solidez de la misma, pero en conjunto con la flexibilidad para no dejar de producir encanto y atracción.

3.2. Diseño de prototipo de App Draw-Hánzí, con base, a las directrices de la metodología ágil **Mobile-D™**.

Para diseñar el prototipo de aplicación móvil, Draw-Hánzí se utilizó la metodología ágil Mobile-D™, que consta de cinco fases de desarrollo como se puede observar en la Figura 54., en la cual se ejemplifican de una manera general los procesos y tiempos de la misma.

Figura 54. Diagrama de procesos de la metodología ágil de desarrollo de software para aplicaciones móviles Mobile-D™.



Autor: Elaboración propia, 2017.

3.2.1. Necesidades previas fundamentales de formación para implementar correctamente **Mobile-D™**, en un desarrollo de software móvil.

Las necesidades de formación necesarias para la correcta implementación de la metodología de desarrollo de software para aplicaciones móviles Mobile-D™, son las siguientes:

- Filosofía de producción y del proceso *Mobile-D™*.
- Ingeniería de requisitos y comunicación con el cliente.
- Planificación de proyectos y estimación de recursos.
- Métricas y SQA.
- Introducción al desarrollo orientado a pruebas (TTD).
- Introducción a SCM.
- Técnica de programación en pares.
- Recopilación automática de métricas.
- Desarrollo orientado a pruebas, prácticas de SCM y métricas.
- Gestión de proyectos, historias, tareas, criterios de salida y prácticas de validación.
- Funciones y responsabilidades, con las plantillas de desglose de trabajo (EDT).
- Pruebas de pre-lanzamiento y lanzamiento.
- Ceremonias de liberación.
- Implementación de la arquitectura móvil.
- Documentación del sistema y del diseño.
- Técnicas de taller post-iteración.
- Planificación de prácticas diurnas.
- Uso y generación de diagramas BPM ó UML.
- Introducción al sistema a desarrollar.

3.2.2. Metodología ágil *Mobile-D™* y sus respectivas fases en el proceso de desarrollo del prototipo de App: Draw-HánzÍ.

El objetivo de esta metodología, es conseguir ciclos de desarrollo muy rápidos, en equipos de desarrollo relativamente muy pequeños. Con base a las metodologías conocidas, *XP*, *CC* y *RUP*.

Se compone en lo general de las siguientes cinco fases:

1. Exploración.
2. Inicialización.
3. Producción.
4. Estabilización
5. Pruebas y corrección de errores.

3.2.2.1. Fase de Exploración

Se centra en la planificación y en los conceptos básicos del proyecto. En esta fase es donde se hace la definición del alcance del proyecto y el establecimiento de las funcionalidades a las que se desea llegar.

Las entradas en esta fase de exploración son:

1. Propuesta (idea) del producto.
2. Conocimiento y uso de la biblioteca de procesos Mobile-D TM.
3. Conocimiento y uso de la biblioteca de procesos organizativos.
4. Contrato.
5. Documento de requisitos iniciales.
6. Plan del proyecto.
7. Normas, restricciones y políticas de la empresa. (en caso de que existan).

Para este punto se da certeza y garantía de que el equipo desarrollador del presente proyecto de tesis, cuenta ya con el conocimiento previo y precisó del inciso 2, de la lista anterior. Sin embargo, por las características propias y particulares del proceso de desarrollo del prototipo de App *Draw-Hánzí*, que se desglosaran en lo sucesivo, se podrá observar que el inciso 3 y 7 quedan descartados, debido a que no se trabaja con un cliente externo (empresa y/o institución), al equipo de desarrollo. En este caso particular, el equipo desarrollador y para fines prácticos, fungirá como cliente y/o desarrollador en lo sucesivo.

Nota 4: Se recalca que, para fines prácticos de la presente investigación científica tecnológica de tesis, los autores toman la decisión de crear dos empresas una “desarrolladora” y una “cliente- empleador”, de forma ficticia, única y exclusivamente para ejemplificar el proceso de desarrollo y demostrar el correcto uso de la metodología ágil Mobile-D TM .

3.2.2.1.1. Propuesta inicial del producto.

Para usos prácticos y como forma de ejemplo, con la finalidad de observar la aplicación de la metodología *Mobile-D* TM, en un contexto lo más parecido a lo real. Se formulará una propuesta ficticia de petición de desarrollo, para el caso de estudio del prototipo de App

Draw-Hánzí, siendo inicialmente la siguiente situación, ilustrada en las Tablas 14 y 15, así como en la Figura 55.

Donde:

Tabla 14. Composición de la empresa Desarrolladora de Software Freelance.

EMPRESA DESARROLLADORA DE SOFTWARE	
Nombre	Puesto
Sandy Trejo Quintero	Dueña
Ulises Moreno Gorgonio	Dueño
Nombre de la empresa	
SUMTSOFT	

Fuente: Elaboración propia, 2017.

Tabla 15. Composición del cliente.

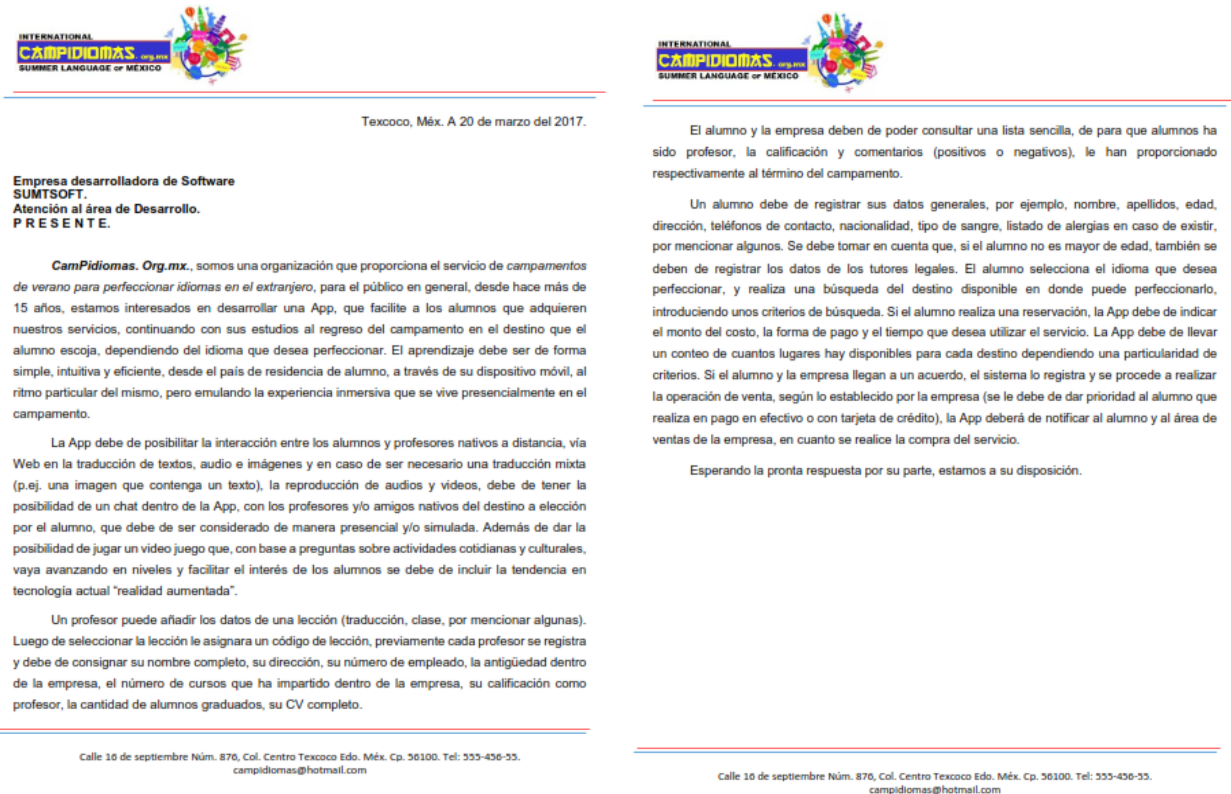
EMPRESA QUE FUNGIRÁ COMO CLIENTE			
Nombre	Puesto	Nivel de Decisión	Alias
Sandy Trejo Quintero	Dueña	100%	Cliente “AT “y Cliente “AQ”.
Ulises Moreno Gorgonio	Dueño	100%	Cliente “BM” y Cliente “BG”.
Nombre de la empresa		Para evitar futuras confusiones se usarán alias a forma de diferenciar hipotéticamente a las personas.	
CamPidiomas.Org.mx			

NOTA 5: No es lo más recomendable ser juez y parte en el desarrollo, ya que se puede interpretar como un sesgo importante en el resultado de la investigación. Pero si se toma en cuenta la creación de un producto privado de la empresa desarrolladora, se daría este tipo de situación exactamente.

Fuente: Elaboración propia, 2017.

A las oficinas de la empresa de desarrollo de software **SUMTSOFT**, llega la siguiente propuesta de trabajo, para el desarrollo de una App.

Figura 55. Carta de petición de desarrollo de App.



Autor: Elaborado por la organización CamPidiomas.Org.mx., 2017.

3.2.2.1.2. Contrato del desarrollo de una aplicación móvil.

La importancia de tener un contrato y llegar a un acuerdo de firma del mismo, yace en la prioridad de *protección*, ya sea de la idea y/o del mismo software que se desarrollará, además de considerar la privacidad para ambas partes desarrollador y/o empresa (cliente).

No obstante, para efectos prácticos del presente proyecto de tesis y del prototipo de App *Draw-HánzÍ*, no se requiere un contrato, por lo cual este punto no se considera. Sin embargo, para emplear el "Manual de Desarrollo de Apps en México", sustentado con la metodología Mobile-DTM, si se muestra un ejemplo, sin valor legal; únicamente como guía de constitución (véase, Anexo II).

3.2.2.2. Etapas de la fase de exploración.

3.2.2.2.1. Establecimiento del proyecto.

Se determina después de la primera reunión con los clientes y próximos a firmar el contrato (en caso de que fuera necesario), que sale fuera del presupuesto económico y del tiempo en que lo necesitan, la petición inicial. Por lo que se establece, como desarrollo una contra oferta ilustrada en la Tabla 16., (solo un módulo de los múltiples, que requiere el desarrollo completo), quedando con la siguiente información:

A. Propuesta temporal del producto

Tabla 16. Especificaciones del producto

PROPUESTA GENERAL DEL 1ER. MÓDULO DE LA APP
<p>La empresa CamPidiomas.Org.mx, requiere del desarrollo, que se encuentre dentro de su presupuesto, que esté totalmente desarrollado y en funcionamiento a finales del 2017. La App, deberá fungir como herramienta interactiva, para el correcto aprendizaje del idioma chino (debido a la creciente demanda del mismo, conjuntamente es el idioma con mayor complejidad en su escritura por lo que requieren de un instrumento facilitador).</p>
<p>Tiene que ser una App sencilla de utilizar, tanto dentro de los campamentos (empresa), como fuera de estos (domicilio de los estudiantes). Sin embargo, también debe de poderse mostrar en las instalaciones de la empresa, para cuando los padres solicitan una clase de prueba (servicio que proporciona la empresa, dentro de su corporativo). Se requiere que la App, funcione para tablets y teléfonos celulares, ya sea de los estudiantes, personal de la empresa y/o profesores.</p>
<p>Deberá mostrar el correcto sentido de la escritura de los caracteres chinos; debe de contener un vocabulario sencillo de palabras (conjunto de caracteres). La App debe de mostrar su correcta pronunciación en lenguaje nativo chino mandarín (pinyin) y la traducción debe de mostrar una imagen representativa universalmente y/o estandarizada del significado de cada palabra en tiempo real (en ningún momento debe de mostrarse traducción alguna en ningún otro idioma que no sea el chino, ya que de esta forma se tratará de emular la experiencia inmersiva que ofrecen sus campamentos, todo el tiempo). Este punto deberá de estar basado en el uso de las nuevas tendencias tecnológicas, dígame, realidad aumentada o realidad virtual y hacerla atractiva para el estudiante. Además de mostrar competitividad ante los competidores de la empresa.</p>
<p>Los profesores deberán de ser capaces de cargar el contenido de las lecciones a la App, de forma fácil y rápida. Esta información deberá de actualizarse de forma automática. Sin embargo, no todos los estudiantes tendrán acceso al contenido total, se les ira desbloqueando según su avance y correspondiente pago.</p>
<p>La App, debe de ser intuitiva, debe tener apariencia educativa, además de contener el logotipo de la empresa, de la lección, así mismo el nombre acordado, así como adjudicar la propiedad a la empresa debidamente. La App deberá funcionar con y sin Internet. Se debe de poder adquirir en una AppStore por el precio indicado y los datos del comprador deberán ser guardados en una base de datos.</p>

Fuente: Elaboración Propia, 2017.

3.2.2.2.2. Establecimiento de las partes interesadas (Stakeholders).

El propósito de esta etapa es identificar y establecer los grupos de interés que se necesitan en las diversas tareas de la fase de exploración, así como en el apoyo de todas las

actividades durante el proceso de desarrollo del proyecto de software. Es necesaria una gran variedad de conocimientos y la cooperación para planificar una aplicación controlada y efectiva del producto de software.

A. *Establecimiento de clientes*

El propósito de esta tarea es establecer el grupo de interés del cliente que tiene el conocimiento del dominio y autoridad sobre los requisitos para el producto de software.

Los objetivos del establecimiento del cliente son:

1. Identificar al cliente(s) participativo para el producto.
2. Ganar el compromiso del cliente, identificado por la participación en el proyecto.
3. Definir el modo (por ejemplo, off / on-site), tareas, funciones y responsabilidades del grupo del cliente.

Para el caso en particular, del desarrollo del prototipo de aplicación móvil Draw Hanzí, se definen e ilustran en la Tabla 17., respectivamente a las partes interesadas centrales de la siguiente manera:

Tabla 17. Definición de las partes interesadas por parte del cliente.

GRUPO DE CLIENTES			
Núm.	Puesto	Alias	Funciones y Responsabilidades
1	Dueño	Cliente AT	<ul style="list-style-type: none"> • Toma de decisiones. • Identificación de requerimientos iniciales. • Priorización de requerimientos finales.
2	Dueño	Cliente BM	<ul style="list-style-type: none"> • Toma de decisiones. • Identificación de requerimientos iniciales. • Priorización de requerimientos finales.
3	Profesor	Cliente AQ	<ul style="list-style-type: none"> • Evaluación y aceptación de las versiones del producto. • Sugerencia de cambios.
4	Personal de Ventas	Cliente BG	<ul style="list-style-type: none"> • Sugerencia de cambios. • Priorización de requerimientos.

Fuente: Elaboración propia, 2017.

B. Establecimiento de parte de los desarrolladores

El propósito de esta tarea es establecer el grupo de interesados, ilustrados en forma en la Tabla 18., por parte de los desarrolladores que tiene el conocimiento del dominio y autoridad sobre el desarrollo del prototipo de App Draw-HánzÍ.

Tabla 18. Definición de las partes interesadas por parte de la empresa de desarrollo de software **SUMTSOFT**.

GRUPO DE SARROLLADOR				
EQUIPO DE EXPLORACIÓN	Nombre del responsable(s)		Alias	
	Sandy Trejo Quintero		Explorador DAAdT	
	Ulises Moreno Gorgonio		Explorador DBdM	
	Funciones y Responsabilidades			
	Este equipo será el encargado de tener el primer encuentro con el cliente y definir las normas generales del proyecto de desarrollo del producto.			
Observaciones				
Aunque no es lo más óptimo, debido a que la metodología Mobile-D™, sugiere que este grupo sea ajeno al grupo de desarrollo.				
Alias		Nombre del responsable(s)		GRUPO DE DIRECCIÓN
Cliente AT		Sandy Trejo Quintero		
Explorador DAAdT				
Cliente BM		Ulises Moreno Gorgonio		
Explorador DBdM				
Funciones y Responsabilidades				
Este grupo tendrá la autoridad para tomar las decisiones y monitorear el progreso del proyecto.				
GRUPO DE SOPORTE	Nombres del responsable(s)		Alias	
	Ing. Nancy Brito Dominguez		Apoyo NAb	
	Ing. Elizur Moreno Gorgonio		Apoyo EIM	
	Funciones y Responsabilidades			
	<ul style="list-style-type: none"> • Apoyo y soporte en la correcta interpretación de la metodología Mobile-D™. • Contribuir a la adaptación y mejora de procesos de software, dentro del equipo desarrollador. 			
Observaciones				
Se define la especialidad de los ingenieros en metodologías de desarrollo de software. Así como en la dirección de proyectos en IT, en México. Han tenido una importante participación en la determinación de metodologías estándares y de escala global dentro de su vasta experiencia a lo largo de su vida laboral. Si bien no conocen a detalle la metodología Mobile-D™, si pueden con base, a su experiencia extrapolar en otras metodologías y casos que dominan a la perfección.				
Se les reconoce la especialización en procesos destinados a contribuir a la adaptación y mejora de procesos y la experiencia en la formación de personal capacitado.				
Funciones y Responsabilidades				
Por el momento no se prevén Testers, ni usuarios finales, ni rutinarlos para el prototipo de App, dentro del desarrollo del presente proyecto de tesis.				
Sin embargo, en algún punto del mismo desarrollo (en el lanzamiento), se acercará el prototipo de App a usuarios y/o testers de oportunidad para documentar y observar algunos puntos de vista (externos al equipo de desarrollo) diferentes sobre la calidad del producto, pero sin publicar el prototipo de App.				
EQUIPO DE SARROLLADOR DEL PROYECTO	Nombre del responsable	Puesto	Alias	Funciones y Responsabilidades
	Sandy Trejo Quintero	Arquitecto	Arq\$T	<ul style="list-style-type: none"> • Orientación y construcción del proyecto. • Adecuado seguimiento del proceso de desarrollo.
		Moderador	Mod\$Q	
		Tester	Test\$S	
		Programador	Prog\$S	
	Ambo	Jefe de Proyecto	JDPamb	
	Ulises Moreno Gorgonio	Diseñador	DisUM	
		Programador	ProgGU	
		Tester	TestUM	
		Auditor Externo	AudTEGU	

Fuente: Elaboración propia, 2017.

3.2.2.2.3. Definición del alcance del proyecto

El propósito de esta etapa es definir las metas para el inicio del proyecto, con respecto al contenido, así como al cronograma del mismo producto de software. En esta etapa del desarrollo, se utilizará el software de administración de proyectos “Microsoft Office Project 2016”.

Los objetivos de la definición del alcance son:

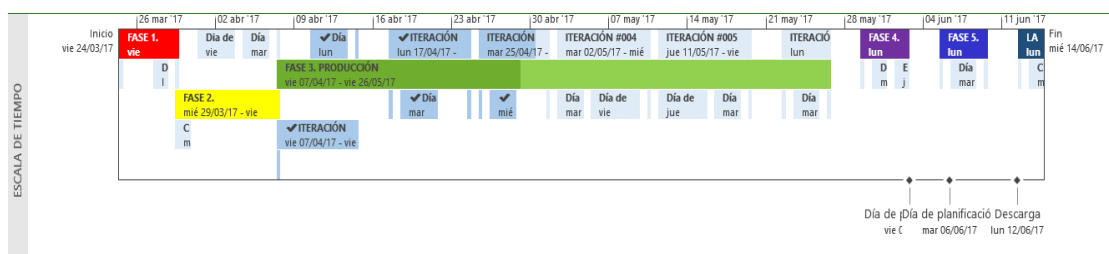
1. Definir el cronograma, para el proyecto de desarrollo. ***
2. Definir los requerimientos iniciales para el producto de software.

Nota: 6: *** Si se desea consultar el cronograma general del presente proyecto de tesis, véase el Anexo V.

A) Planificación inicial del proyecto

1. **Definición de la línea de tiempo:** en este paso se define e ilustra en la Figura 56., el calendario general estimado para el prototipo de App.

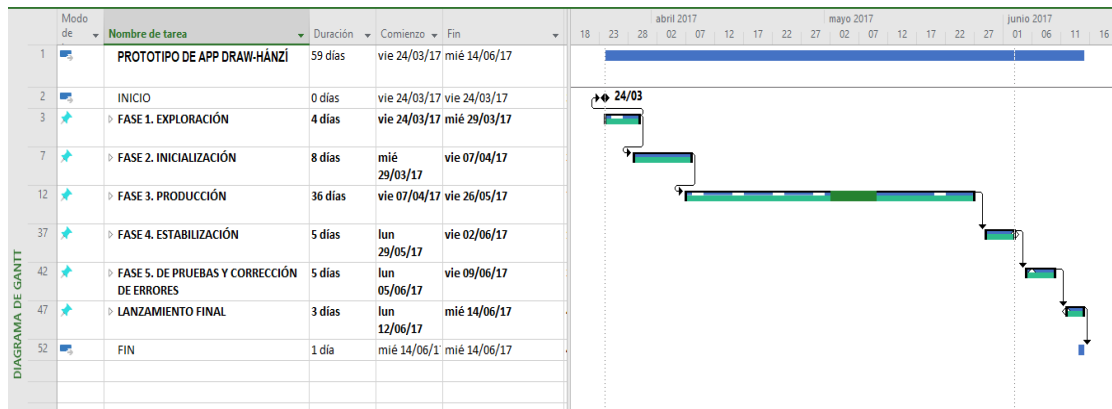
Figura 56. Línea del tiempo del proyecto



Autor: Elaboración Propia, 2017.

2. **Definir el ritmo:** en este paso se estima un número siete iteraciones, con una longitud de entre 7 a 5 días cada una. Se definen e ilustran en la Figura 57., diez entregas de artefactos, aproximadamente dentro del desarrollo del prototipo de App: Draw-Hánzí.

Figura 57. Definición del ritmo del proyecto del prototipo de App.



Fuente: Elaboración propia, 2017.

3. **Definir / estimar las inversiones:** en este paso se acuerdan los costos del proyecto y el esfuerzo que implicara la creación del prototipo de App, ilustrado respectivamente en las desde la Tabla 19 hasta la Tabla 26.

Tabla 19. Estimación inicial de horas (esfuerzo)

FASES	RESPONSABLE	HORAS
FASE 1. Exploración	Jefe de Proyecto	32
	Exploradores	
FASE 2. Inicialización	Jefe de Proyecto	64
	Exploradores	
	Arquitecto	
	Diseñador	
FASE 3. Producción	Analista	288
	Diseñador	
	70% Programadores	
	30% Testers	
FASE 4. Estabilización	Analista	40
	Diseñador	
	50% Programador	
	50% Testers	
FASE 5. De pruebas y corrección de errores	50% Analista	40
	50% Diseñador	
	40% Programador	
	60% Testers	
Lanzamiento Final	Jefe de Proyecto	24
	Exploradores	
TOTAL DE HORAS (Esfuerzo)		488

Fuente: Elaboración propia, 2017.

Tabla 20. Identificación y estimación de costos humanos iniciales

Costos humanos iniciales					
La remuneración utilizada para esta estimación corresponde a un promedio mínimo y máximo, del precio por hora que se obtiene de diversas investigaciones informales de empleados en IT, Se considera que el año tiene 250 días laborales.					
Rol	Horas Estimadas	Remuneración (\$/hrs ^{***}) (Caso 1)	Remuneración (\$/hrs ^{***}) (Caso 2)	Costo Estimado \$ (Caso 1)	Costo Estimado \$ (Caso 2)
Jefe de Proyecto	28	\$ 50.47	\$ 282.66	\$1,413.16	\$ 7,914.48
Explorador	44	\$ 50.47	\$ 262.47	\$ 2,220.68	\$ 11,548.68
Analista	108	\$ 50.47	\$ 226.12	\$ 5,450.76	\$ 24,420.96
Arquitecto	16	\$ 50.47	\$ 226.12	\$ 807.52	\$ 3,617.92
Diseñador	108	\$ 50.47	\$ 226.12	\$ 5,450.76	\$ 24,420.96
Programador	92	\$ 50.47	\$ 226.12	\$ 4,643.24	\$ 20,803.04
Tester	92	\$ 50.47	\$ 155.46	\$ 4,643.24	\$ 14,302.32
TOTAL	488			\$ 24,629.96	\$ 107,028.36

Fuente: Elaboración propia, 2017.

Tabla 21. Definición de recursos materiales

RECURSOS	TIPO	FINALIDAD
Computadora DELL Inspiron N5050 Core i3 de 4GB con Windows 10	Herramienta de desarrollo	Para desarrollar la App y Base de Datos
Computadora DELL Inspiron N5110 Core i7 de 6GB con Windows 7	Herramienta de desarrollo	Para desarrollar la App y Base de Datos
Computadora HP Intel Celeron de 4GB con Windows 10	Herramienta de desarrollo	Para desarrollar la App y Base de Datos
Motorola G4 RAM 2G, 16GB Memoria interna, con cámara 13Mpx.	Herramienta de desarrollo	Para probar la App, en OS Android
Zumm P50 RAM 1G , 16GB Memoria interna, con cámara de 13Mpx.	Herramienta de desarrollo	Para probar la App, en OS Android
Microsoft Office Word 2016 Microsoft Office PowerPoint 2016	Herramienta de desarrollo	Para realizar la documentación del Proyecto
Motor UNITY 3D	Herramienta de desarrollo	Permitirá construir App basadas en AR
Vuforia	Herramienta de desarrollo	Permitirá construir App basadas en AR
Android Studio	Herramienta de desarrollo	Para desarrollar Apps basadas en OS Android
SublimeText	Herramienta de desarrollo	Editor de texto para desarrollar la App
Cacoo	Herramienta de desarrollo	Editor de Mockups para desarrollar la App
Microsoft Office Project 2016	Herramienta de desarrollo	Para planificar el Proyecto
NitroPro 10	Herramienta de desarrollo	Para visualizar documentos
DropBox Server Drive Server	Herramienta de desarrollo	Para el control de versiones del repositorio del Código Fuente
Correo Electronico Gmail	Herramienta de comunicación	Para comunicación entre el equipo desarrollador y el cliente

Autor: Elaboración propia, 2017.

Tabla 22. Identificación y costos indirectos de hardware

COSTOS DE HARDWARE				
Se considera que los recursos de hardware que se usarán, se estima que tendrán una vida útil de 4 años. Para establecer un costo de amortización por hora de uso se toma de referencia que 1 año de vida útil tiene 250 días laborales a un ritmo de trabajo de 8 horas diarias.				
Para realizar el cálculo de la amortización se hará de la siguiente manera:				
Precio				
4 años (250 días laborales * 8 horas al día)				
Recursos Hardware	Precio (\$)	Horas Estimadas	Costos de amortización (\$/h)	Costo estimado (\$)
Computadora DELL Inspiron N5050 Core i3 de 4GB con Windows 10	\$ 10,990.-	110	\$ 1.37	\$ 150.70
Computadora DELL Inspiron N5110 Core i7 de 6GB con Windows 7	\$ 22,799.-	109	\$ 2.84	\$ 309.56
Computadora HP Intel Celeron de 4GB con Windows 10	\$ 7,867.-	110	\$ 0.98	\$ 107.80
Motorola G4 RAM 2G, 16GB Memoria interna, con cámara 13Mpx.	\$ 4,799.-	80	\$ 0.59	\$ 47.20
Zumm P50 RAM 1G, 16GB Memoria interna, con cámara de 13Mpx.	\$ 4,950.-	79	\$ 0.61	\$ 48.19
TOTAL				\$ 663.45

Fuente: Elaboración propia, 2017.

Tabla 23. Identificación de costos indirectos de software

COSTOS DE SOFTWARE				
Se considera que los recursos de software que se usarán, se estima que tendrán una vida útil de 3 años. Para establecer un costo de amortización por hora de uso se toma de referencia que 1 año de vida útil tiene 250 días laborales a un ritmo de trabajo de 8 horas diarias.				
Para realizar el cálculo de la amortización se hará de la siguiente manera:				
Precio				
3 años (250 días laborales * 8 horas al día)				
<ul style="list-style-type: none"> No se ha tomado en cuenta el costo del servidor (Azure App Service), ya que se cuenta con una versión gratuita. 				
Recursos Software	Precio (\$)	Horas Estimadas	Costos de amortización (\$/h)	Costo estimado (\$)
Microsoft Office Word 2016	\$ 1,299.-	100	\$ 0.21	\$ 21.-
Microsoft Office PowerPoint 2016				
Microsoft Office Project 2016				
Motor UNITY 3D	\$ 7,341.60	388	\$ 1.22	\$ 473.36
Vuforia	\$ 20,766.24	388	\$ 3.46	\$ 1,342.48
Android Studio	0	No Aplicá	0	
SublimeText	0	No Aplicá	0	
Cacoo	0	No Aplicá	0	
NitroPro 10	0	No Aplicá	0	
DropBox Server	0	No Aplicá	0	
Drive Server				
Correo Electronico Gmail	0	No Aplicá	0	
Microsoft Windows 10	0	No Aplicá	0	
Microsoft Windows 7				
TOTAL				\$ 1,836.84

Fuente: Elaboración propia, 2017.

Tabla 24. Identificación y estimación de costos generales

COSTOS GENERALES			
Durante el transcurso del proyecto de creación del prototipo de App, se estará utilizando los equipos de cómputo antes mencionados, lo que generará un consumo energético. También se ha de tener en cuenta la recarga de la batería de los 2 Smartphones que se utilizarán (se considera que cada uno se tiene que recargar 2 veces al día y durante máximo 3 horas). Respecto a la conexión de Internet, se utilizará una banda ancha de 10 GB. También se considerará que hipotéticamente se hará un desplazamiento en transporte público en el área metropolitana del Estado de México, utilizando las rutas correspondientes.			
Recurso	Precio	Cantidad	Costo estimado (\$)
Consumo Energético	\$ 0.978 / kWh	Consumo diario de 1.8 kWh durante 4 meses	\$ 214.76
Conexión de Internet	\$ 538.- / Mensuales	4 meses	\$ 2,152.-
Desplazamientos	\$ 1,800.- cada 3 meses	4 meses	\$ 2,400.-
Costo por publicación en AppStore	\$650.- Pago único	No Aplica	\$ 650.-
TOTAL			\$ 5,416.76

Fuente: Elaboración Propia, 2017.

Tabla 25. Identificación y estimación de costos de contingencia

COSTOS DE CONTINGENCIA		
Como medida de contingencia se establece un margen del 20% de la suma de los costos humanos, costos de hardware y software. Se considerará el 10% de los costos directos para imprevistos.		
Recurso	Costo (\$) (Caso 1)	Costo (\$) (Caso 2)
Costos humanos	\$ 24,629.96	\$ 107,028.36
Costos de hardware	\$ 663.45	\$ 663.45
Costos de Software	\$ 1,836.84	\$ 1,836.84
SUBTOTAL 1	\$ 27,130.25	\$ 109,528.65
SUBTOTAL 2	\$ 5,426.05	\$ 21,905.73
Imprevistos	\$ 2,462.99	\$ 10,702.83
TOTAL	\$ 7,889.04	\$ 32,608.56

Fuente: Elaboración propia, 2017.

Tabla 26. Identificación y estimación del costo total

COSTOS TOTAL		
Como se puede observar en el costo total para cada caso que se planificó durante la fase inicial. Para realizar el cálculo final se deberá tener en cuenta lo siguiente: No se llevará a cabo ningún aumento de precio durante el proyecto, ya que no es de una duración muy extensa. Dado que es un proyecto sin ánimo de lucro, no se añadirá al presupuesto final ninguna ganancia (beneficio) sobre el costo total.		
Recurso	Costo (\$) (Caso 1)	Costo (\$) (Caso 2)
Costos humanos	\$ 24,629.96	\$ 107,028.36
Costos de hardware	\$ 663.45	\$ 663.45
Costos de software	\$ 1,836.84	\$ 1,836.84
Costos generales	\$ 5,416.76	\$ 5,416.76
Costos de contingencia	\$ 7,889.04	\$ 32,608.56
TOTAL	\$ 40,436.05	\$ 147,553.97

Fuente: Elaboración propia, 2017.

B) Colección de requisitos iniciales

A continuación, se van a describir los requisitos iniciales, tanto funcionales, como no funcionales, ilustrados en la Tabla 27., que se han tenido en cuenta para la creación el prototipo de App y que esta sea considerada con una alta calidad.

Tabla 27. Requerimientos generales.

REQUERIMIENTOS DE LA APP, TOMADOS DE LA ESPECIFICACIÓN DEL PRODUCTO	
Requerimiento General	
1.	La App está en funcionamiento pleno en el mes de Agosto del 2017. Es decir, Deberá estar instalada en todos los dispositivos empleados en el corporativo de la empresa, para esta fecha.
Requerimientos Funcionales	
1.	La App deberá mostrar vía vídeo, el correcto sentido de la escritura (caligrafía) de los caracteres chinos del vocabulario, que se proporcione a discreción de la empresa.
2.	La App contendrá un vocabulario de caracteres, inicial proporcionado a discreción por la empresa.
3.	La App deberá demostrar la correcta pronunciación vía audio, en el tono nativo en chino mandarín (pinyin), cada uno de los caracteres.
4.	La App deberá traducir los caracteres chinos del vocabulario, con una imagen estándar de su significado en tiempo real. No obstante, en ningún momento deberá ofrecer ninguna otra traducción a otro idioma que no sea el chino.
5.	La App deberá de ofrecer una experiencia apropiada de las nuevas tendencias tecnológicas como son la realidad aumentada.
6.	Los profesores deberán de ser capaces de cargar el contenido de las lecciones de manera sencilla y en menos de 5 minutos y desde cualquier lugar en donde se encuentren.
7.	La información de las lecciones nuevas, que los profesores carguen en la App deberá de actualizarse de forma automática en todas las App que los estudiantes ya descargaron.
8.	Restringir el Acceso de los Estudiantes no autorizados al contenido total de la herramienta. Es decir, se les ira desbloqueando lecciones y contenido, dependiendo de su avance en las lecciones y a su correspondiente pago por lección.
9.	Los datos de los estudiantes (comprador), deberá ser guardado en una base de datos, de fácil consulta para la empresa.
Requerimientos NO Funcionales	
10.	Un usuario experimentado debe ser capaz de utilizar todas las funciones del sistema, tras un entrenamiento máximo de 2 horas, tras el cual no cometerá más de 2 errores diarios en promedio.
11.	La App debe de tener un correcto funcionamiento en los dispositivos con sistema operativo Android.
12.	La App debe de tener un correcto funcionamiento responsivo para ser compatible con los múltiples tamaños de pantallas de las Tablets y Smartphones.
13.	La App debe de ser intuitiva. Es decir, no deberá de requerir explicación alguna sobre el contenido estético de la misma.
14.	La App deberá tener apariencia educativa desde el inicio.
15.	La App deberá contener el logotipo de la empresa a manera de adjudicar la debida propiedad de la misma.
16.	La App deberá de contener el logotipo de la lección a elegir.
17.	La App deberá de funcionar correctamente con y sin necesaria la conexión a Internet. Es decir, se debe de realizar un respaldo del avance del alumno en caso de ser necesario y actualizarse en cuanto de conecte a una red.
18.	La App deberá estar disponible en la AppStore (Google Play), al precio que la empresa considere.

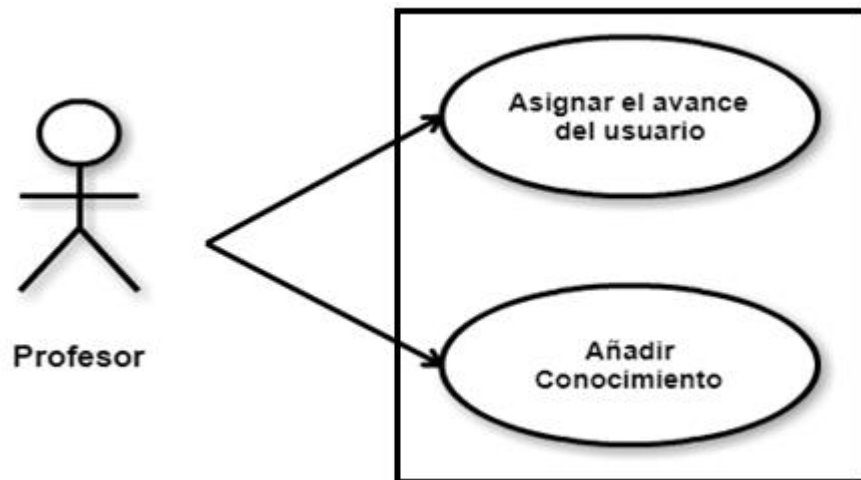
Fuente: Elaboración Propia, 2017.

• Diagramas generales de casos de uso para el prototipo de App Draw-Hánzǐ

A continuación, se presentan gráficamente en la Figura 58, 59 y 60, así como las desde la Tabla 28 hasta la Tabla 40, respectivamente, los diagramas de casos de uso⁸ del prototipo de App, relacionándolos con los actores implicados.

⁸ Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa de los diagramas UML, compuesto por diversos elementos gráficos que se combinan para conformar diagramas; es una técnica de aciertos y errores para obtener los requerimientos de la App, desde el punto de vista del usuario. Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

Figura 58. Diagrama de casos de uso del profesor



Fuente: Elaboración Propia, 2017.

Tabla 28. Descripción del caso de uso “Asignar el avance del usuario”

ASIGNAR EL AVANCE DEL USUARIO	
Actor principal	Profesor
Disparador	El profesor determina en que punto se puede acceder a otra lección, a manera de avance del curso.
Precondiciones	
El profesor asigna las acciones y secciones que corresponden a una lección. Debido a un permiso que le permite modificar el contenido en el servidor.	
Escenario principal del éxito	
<ul style="list-style-type: none"> . El profesor inicia sesión en la consola del servidor. . Modifica el fichero del servidor que contiene los permisos, que se muestran en la App. . Sube al servidor el fichero actualizado. 	

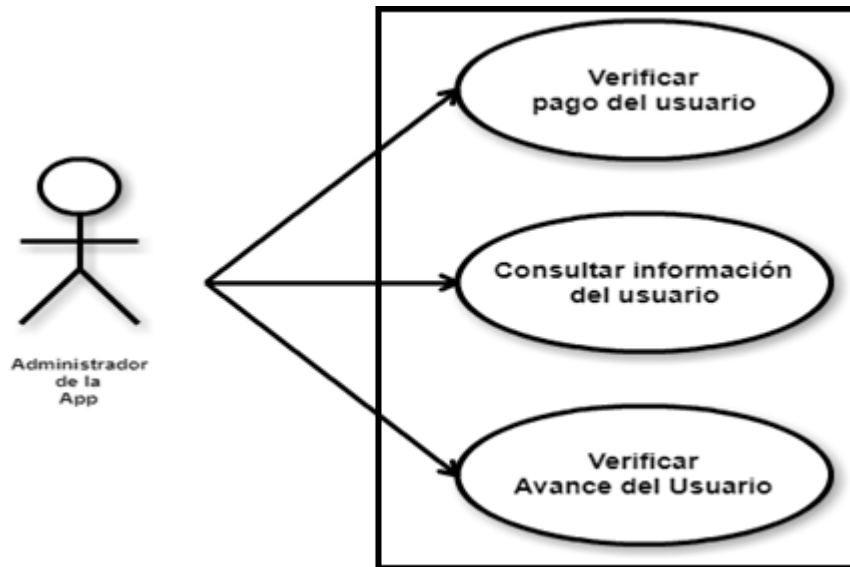
Fuente: Elaboración propia, 2017.

Tabla 29. Descripción del caso de uso “Añadir conocimiento”

AÑADIR CONOCIMIENTO	
Actor principal	Profesor
Disparador	El profesor quiere añadir conocimiento a la App
Precondiciones	
El profesor tiene permisos para modificar el contenido desde el servidor.	
Escenario principal del éxito	
<ul style="list-style-type: none"> . El profesor inicia sesión en la consola del servidor. . Modifica el fichero del servidor que contiene todas las lecciones que se muestran en la App. . Sube al servidor el fichero actualizado. 	

Fuente: Elaboración propia, 2017.

Figura 59. Diagrama de casos de uso del Administrador



Autor: Elaboración Propia, 2017.

Tabla 30. Descripción del caso de uso “Verificar pago del usuario”

VERIFICAR EL PAGO DEL USUARIO	
Actor principal	Administrador de la App
Disparador	El administrador quiere verificar que todos los usuarios han realizado el pago correspondiente.
Precondiciones	
El administrador tiene permisos para verificar el contenido de la BD en el servidor y deshabilitar el funcionamiento de la App, vía remota.	
Escenario principal del éxito	
. El administrador inicia sesión en la consola del servidor.	
. Accede al contenedor que desea de la BD.	
. Accede a la tabla que desea y verifica la información de pago.	

Fuente: Elaboración propia, 2017.

Tabla 31. Descripción del caso de uso “Consultar información de usuario”

CONSULTAR INFORMACIÓN DE USUARIO	
Actor principal	Administrador de la App
Disparador	El administrador quiere consultar la información de un usuario de la BD.
Precondiciones	
El administrador tiene los permisos para consultar el contenido de la BD	
Escenario principal del éxito	
. El administrador inicia sesión en la consulta del servidor.	
. Accede al contenedor que desea de la BD.	
. Accede a la tabla que desea y consulta la información del usuario.	

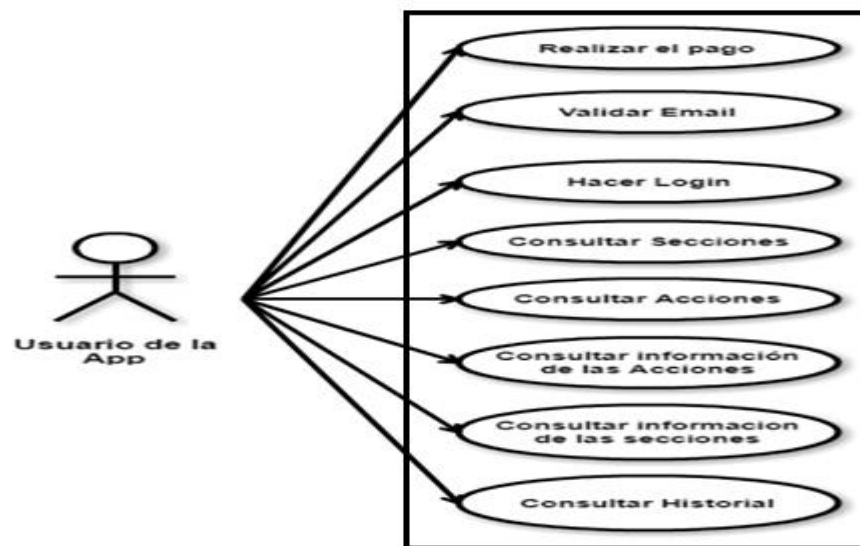
Fuente: Elaboración propia, 2017.

Tabla 32. Descripción del caso de uso “Verificar avance del usuario”

VERIFICAR AVANCE DEL USUARIO	
Actor principal	Administrador de la App
Disparador	El administrador quiere verificar el avance del usuario
Precondiciones	
El administrador tiene permisos para verificar el avance del usuario de la BD en el servidor.	
Escenario principal del éxito	
<ul style="list-style-type: none"> . El administrador inicia sesión en la consola del servidor. . Accede al contenedor que desea de la BD. . Accede a la tabla que desea y verifica el avance del usuario. 	

Fuente: Elaboración propia, 2017.

Figura 60. Diagrama de casos de uso del usuario de la App



Autor: Elaboración Propia, 2017.

Tabla 33. Descripción del caso de uso “Realizar pago”

REALIZAR PAGO	
Actor principal	Usuario de la App
Disparador	El usuario desea descargar la App. de la AppStore, previo pago de pago de la misma.
Precondiciones	
El usuario ingresa a la AppStore, realiza el previo pago, descarga e instala la App.	
Escenario principal del éxito	
<ul style="list-style-type: none"> . El usuario ingresa a la AppStore. . El usuario busca la App en el repositorio de la AppStore. . Realiza el pago pertinente, en la modalidad más conveniente para el usuario. . Descarga e instala la App en su dispositivo. 	

Autor: Elaboración propia, 2017.

Tabla 34. Descripción del caso de uso “Validar Email”

VALIDAR EMAIL	
Actor principal	Usuario de la App
Disparador	El usuario quiere hacer uso de la App
Precondiciones	
El usuario no ha iniciado sesión, ni se ha registrado en la App.	
Escenario principal del éxito	
<ul style="list-style-type: none"> . El usuario abre la App. . La App muestra la pantalla de validación de email. . El usuario hace clic en la casilla correspondiente e introducir el email . La App muestra el teclado del dispositivo. . El usuario escribe el email que quiere validar en la App. . La App valida el email <p>1 Si el email ya existe en la App se realiza el caso de uso “Hacer Login” 2 Si el email no existe en la App se realiza el caso de uso “Validar Email”</p>	

Fuente: Elaboración propia, 2017.

Tabla 35. Descripción del caso de uso “Hacer Login”

HACER LOGIN	
Actor principal	Usuario de la App
Disparador	El usuario quiere iniciar sesión en la App.
Precondiciones	
El usuario ha validado el email y este email ya había sido registrado previamente en la App.	
Escenario principal del éxito	
<ul style="list-style-type: none"> . El usuario hace clic en la casilla correspondiente para introducir la contraseña que corresponde al email, introducido en el paso anterior. . El App muestra el teclado del dispositivo. . El usuario escribe la contraseña que desea que la App valide. . El usuario hace clic en el botón de entrar . El App valida la contraseña. . El App autoriza al usuario para utilizar la misma. 	

Fuente: Elaboración propia, 2017.

Tabla 36. Descripción del caso de uso “Consultar Secciones”

CONSULTAR SECCIONES	
Actor principal	Usuario de la App
Disparador	El usuario quiere consultar las secciones que ofrece la App.
Precondiciones	
El usuario tiene la sesión iniciada en la App y ha abierto la misma.	
Escenario principal del éxito	
<ul style="list-style-type: none"> . El usuario hace clic en la sección “idioma” del menú principal. . La App muestra la pantalla del Idioma, que contiene las secciones que puede consultar el usuario. . El usuario puede consultar las diferentes secciones que la App ofrece. La App ofrece por el momento únicamente una única sección, pero en teoría podría navegar por las siguientes, por ejemplo: Chino Inglés Francés 	

Autor: Elaboración propia, 2017.

Tabla 37. Descripción del caso de uso “Consultar Acciones”

CONSULTAR ACCIONES	
Actor principal	Usuario de la App
Disparador	El usuario quiere consultar las acciones que ofrece la App.
Precondiciones	
El usuario tiene la sesión iniciada en la App y ha abierto la misma.	
Escenario principal del éxito	
<ul style="list-style-type: none"> . El usuario hace clic en la pestaña “Acción” del submenú. . El App muestra la pantalla de la “Acción”, que contienen las secciones a consultar por el usuario. . El usuario navega por las diferentes secciones y selecciona la que le gustaría consultar y probar sus acciones. . El App muestra las acciones de la sección seleccionada por el usuario. . El usuario consulta las diferentes acciones que la App ofrece. El App ofrece tres acciones para cada una de las secciones: Traducción Caligrafía Reconocimiento de patrones 	

Fuente: Elaboración propia, 2017.

Tabla 38. Descripción del caso de uso “Consultar información de las Acciones”

CONSULTAR INFORMACIÓN DE LAS ACCIONES	
Actor principal	Usuario de la App
Disparador	El usuario quiere consultar la información de una acción por el App.
Precondiciones	
El usuario tiene la sesión iniciada en la App y ha abierto la misma.	
Escenario principal del éxito	
<ul style="list-style-type: none"> . El usuario hace clic en la pestaña “Acción” del submenú. . La App muestra la pantalla de la “Acción”, que contienen las secciones a consultar por el usuario. . El usuario navega por las diferentes secciones y selecciona la que le gustaría consultar y probar sus acciones. . El App muestra el botón de “info” de la sección seleccionada junto con las acciones de la sección. . El usuario hace clic en el botón “info”. . El App muestra una pantalla con la información de las acciones. . El usuario consulta la información de la acción. 	

Fuente: Elaboración propia, 2017.

Tabla 39. Descripción del caso de uso “Consultar información de las secciones”

CONSULTAR INFORMACIÓN DE LAS SECCIONES	
Actor principal	Usuario de la App
Disparador	El usuario quiere consultar la información de una acción por el App.
Precondiciones	
El usuario tiene la sesión iniciada en la App y ha abierto la misma.	
Escenario principal del éxito	
<ul style="list-style-type: none"> . El usuario hace clic en la sección “idioma” del menú principal. . La App muestra la pantalla del “Idioma”, que contiene las secciones que puede consultar el usuario. . El usuario navega por las diferentes secciones y selecciona la sección requerida. . El App muestra el botón de “info” de la selección seleccionada junto con las acciones de la sección. . El usuario hace clic en el botón “info”. . El App muestra una pantalla con la información de la sección. . El usuario consulta la información de la sección. 	

Fuente: Elaboración propia, 2017.

Tabla 40. Descripción del caso de uso “Consultar historial”

CONSULTAR HISTORIAL	
Actor principal	Usuario de la App
Disparador	El usuario quiere consultar el historial de sus acciones que el App ofrece.
Precondiciones	
El usuario tiene la sesión iniciada en el sistema y ha abierto la misma.	
Escenario principal del éxito	
<ul style="list-style-type: none"> El usuario hace clic en la pestaña de “Historia” del menú. El App muestra la pantalla “historia”, que contiene las secciones a consultar por el usuario. El usuario navega y consulta el historial de las acciones que ha llevado a cabo hasta el momento. 	

Fuente: Elaboración propia, 2017.

- **Necesidades de formación para el prototipo de App Draw-Hánzi**

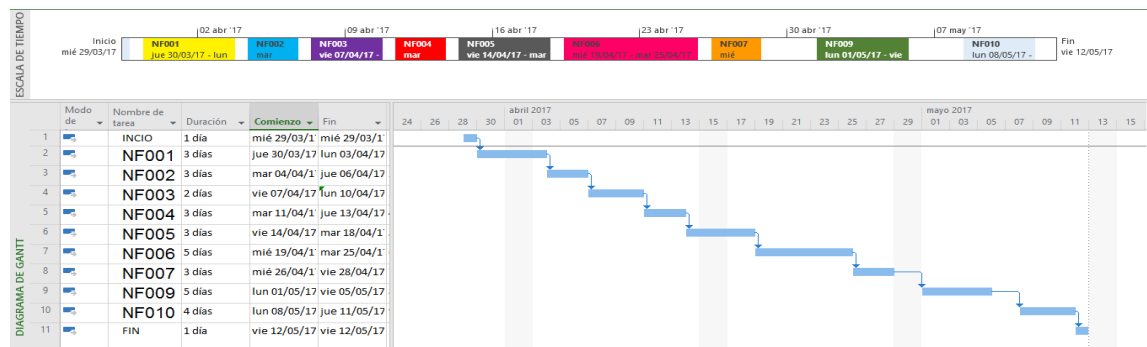
Se estiman las siguientes necesidades de formación para la culminación exitosa del prototipo de App: Draw-Hánzi, mismas que se ilustran en la Tabla 41 y la Figura 58, respectivamente.

Tabla 41. Ejemplo de necesidades de información

NÚMERO DE IDENTIFICACIÓN	NECESIDADES DE FORMACIÓN
NF001	Cómo implementar funcionalidades importantes: pasar ficheros, XML, Acceso a base de datos.
NF002	Cómo hacer para estar presentes en alguna de las principales tiendas de aplicaciones.
NF003	Implementación y mantenimiento del Business Activity Monitor (BAN), monitorización de KPI's y generación de cuadros de mando y alarmas.
NF004	Formación técnica en productos de la suite WSO2.
NF005	Formación para la instalación y configuración de herramientas de la suite WSO2 como: API Manager, ESB, Broker, BPS, DSS, Identity Server, por mencionar algunas
NF006	Consultoría en todo el proceso de adopción de una solución de API Manager.
NF007	Implantación de una solución tecnológica de API Manager.
NF009	Creación de API's REST para recubrir funcionalidad ya existente o crear una nueva.
NF010	Migración de servicios existentes.

Fuente: Elaboración propia, 2017.

Figura 61. Calendario de formación



Fuente: Elaboración propia, 2017.

3.2.2.2.4. Lista de verificación en la fase de exploración

Debido a que se culmina la planificación de la fase de exploración se decide, realizar el CheckList de control y medición, ilustrado en la Tabla 42.

Tabla 42. CheckList del plan proyecto en la fase de exploración

LISTA DE VERIFICACIÓN DEL PLAN PROYECTO EN LA EXPLORACIÓN			
FASE 1. EXPLORACIÓN			
Requisitos Iniciales	SI	NO	N/A
Todos los requisitos funcionales iniciales, se han incluido en el plan del proyecto	X		
Todos los requisitos no funcionales iniciales, se han incluido en el plan del proyecto	X		
Calendario y Ritmo	SI	NO	N/A
El calendario general se ha incluido en el plan del proyecto	X		
El ritmo previsto (número y longitud de cada fase, así como sus iteraciones), han sido definidas en el plan del proyecto	X		
Recursos	SI	NO	N/A
Se han incluido en el plan del proyecto a todos los grupos de interés identificados y sus respectivos miembros (p.ej. grupo de clientes, grupo directivo y grupo de arquitectura)	X		
Se han incluido en el plan del proyecto la información relativa al software seleccionado (p.ej. herramientas de desarrollo, dispositivos, por mencionar algunos)	X		
El plan del proyecto se ha incluido a los miembros del equipo desarrollador identificados	X		
Formación	SI	NO	N/A
Las necesidades de capacitación de equipo se han incluido en el plan del proyecto	X		
El calendario de capacitación ha sido incluido en el plan del proyecto	X		

Fuente: Elaboración propia, 2017.

3.2.2.2 FASE DE INICIACIÓN

El propósito de esta fase de inicialización es permitir el éxito de las siguientes fases del proyecto mediante la preparación y verificación de todas las cuestiones críticas fundamentales del desarrollo, todos están en plena preparación al final de la fase de implementación de los requisitos seleccionados por el cliente.

Los objetivos de esta fase de inicialización son:

1. Obtener una buena comprensión global del producto: por parte del equipo de desarrollo del proyecto, sobre los requisitos iniciales y la línea de la arquitectura.
2. Preparar los requisitos físicos, técnicos y humanos, así como la comunicación con el cliente, los planes del proyecto y todas las cuestiones fundamentales de desarrollo: de manera que los problemas están todos plenamente preparados para la aplicación de los requisitos, durante las próximas fases del proyecto.

Los criterios de entrada de la fase de inicialización son:

1. La fase precedente de exploración se ha completado correctamente.

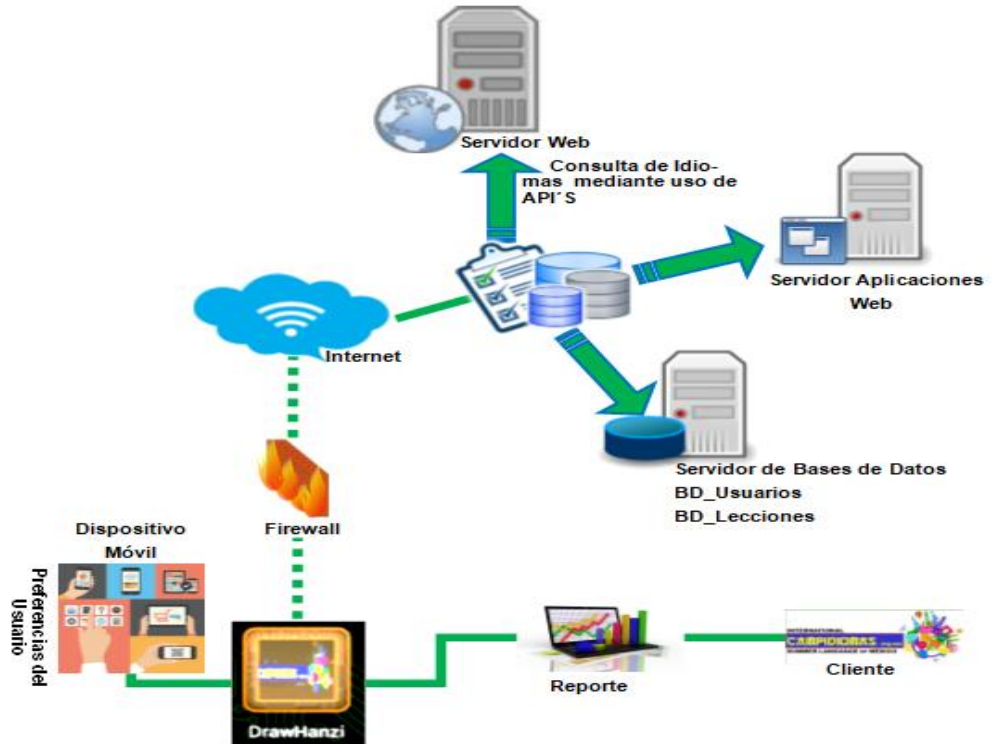
3.2.2.2.1. Configuración del proyecto

A. Arquitectura del prototipo de App Draw-Hánzí

A continuación, se muestra la arquitectura general del prototipo de App, ilustrada en las Figuras 62 y 63., la misma está orientada a servicios y por lo tanto consta de las siguientes tres partes:

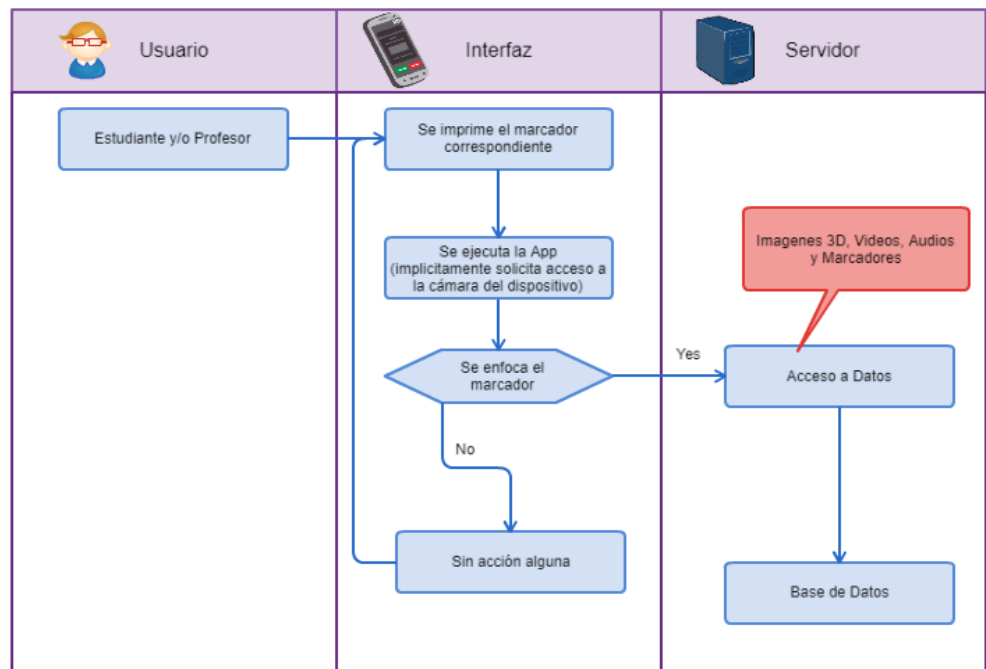
- Aplicación Móvil.
- Servicio Web.
- Servidor de Base de Datos.

Figura 62. Arquitectura grafica del prototipo de App



Fuente: Elaboración propia, 2017.

Figura 63. Arquitectura funcional del prototipo de App



Fuente: Elaboración propia, 2017.

B. Establecimiento de recursos del cliente

A continuación, se determinará en que OS es más conveniente se desarrolle la App y de que versión a que versión deberá abarcar el mismo, para su correcto funcionamiento, dependiendo de los recursos con los que cuenta el cliente, como lo muestra la Tabla 43.

Tabla 43. Recursos físicos del cliente

CANTIDAD	TIPO	RECURSO	SISTEMA OPERATIVO	VERSIÓN	CARACTERÍSTICAS
2	Smartphone	Samsung Galaxy S7	Android	6.0.1	Cámara 12 Mpx.
1	Smartphone	Huawei Gr3	Android	5.1	Cámara de 13 Mpx.
2	Smartphone	Sony Xperia Z5	Android	7.0	Cámara de 20.7 Mpx.
2	Tablet	Connect	Android	5.1	Cámara de 8 Mpx.
1	Tablet	Samsung Galaxy Tab A	Android	6.0.1	Cámara de 13 Mpx.



Nota 7: Resultado de la Tabla 38, se determina que la App deberá ejecutarse correctamente en el OS Android desde la versión 4.2 hasta la 7.0, del mismo.

Fuente: Elaboración propia, 2017.

C. Establecimiento de la comunicación con el cliente

A continuación, se determina con el cliente que no requiere de información extraordinaria a la marcada en el calendario oficial. No obstante, se determina que en caso de ser estrictamente necesario se asignan los días lunes a las 08:00 Hrs. Como horario determinado para contactar con él, con los tipos de comunicación como se determinan en la Tabla 44.

Tabla 44. Definición de comunicación del equipo desarrollador con el cliente

FORMAS DE COMUNICACION ENTRE EL EQUIPO DESARROLLADOR Y EL CLIENTE			
El proceso de comunicación entre el cliente y el equipo desarrollador, se desarrollará a través de las herramientas de comunicación sincrónicas y asincrónicas, bajo las particularidades de cada una de estas, mismas que marcan puntos importantes de conocer y aplicarlas en el desarrollo del producto de software.			
TIPOS DE COMUNICACION			
Sincrónica	Es aquella comunicación que nos permite interacción en tiempo real entre los interesados. Este tipo de comunicación se da en situaciones tales como cuando se conversa con alguien a través del teléfono o cuando se participa en una sesión de videoconferencia o chat.		
CARACTERÍSTICAS DE LA COMUNICACIÓN SINCRÓNICA			
Es independiente del lugar	Es temporalmente dependiente	Se basa en texto	La comunicación tiene lugar en grupo o individual
La comunicación se produce al encontrarse físicamente los interesados ubicados en contextos distintos.	La comunicación no necesariamente requiere que se coincida en el mismo tiempo.	La comunicación se desarrolla fundamentalmente en un formato escrito (textual), aunque también se puede realizar en formato audio y/o visual.	La comunicación se produce en presencia de varios interesados o entre un solo interesado, por ejemplo, un correo electrónico.
EJEMPLOS DE COMUNICACIÓN SINCRÓNICA		SITUACIÓN ESPECIAL	
 <p>Autor: WhatsApp Inc, Facebook, Skype Microsoft, Google Inc, 2017</p>		Hay un punto en específico, para el uso de este tipo de comunicación que irremediablemente tiende a volverse asincrónica, sin remedio alguno.	
Asincrónica	Es aquella comunicación no permite la interacción en tiempo real entre los interesados, en diferentes espacios de tiempo.		
CARACTERÍSTICAS DE COMUNICACIÓN ASINCRÓNICA			
Es independiente del lugar	Es temporalmente independiente	Se basa en texto	La comunicación tiene lugar en grupo o individual
La comunicación se produce al encontrarse físicamente los interesados ubicados en contextos distintos.	La comunicación no necesariamente requiere que se coincida en el mismo tiempo.	La comunicación se desarrolla fundamentalmente en un formato escrito (textual).	La comunicación se produce en presencia de varios interesados o entre un solo interesado, por ejemplo, en foros.
EJEMPLOS DE COMUNICACIÓN ASINCRÓNICA			
 <p>Autor: Google Inc. Facebook, Dropbox Business, Twitter Inc., 2017.</p>			

NOTA 8: En el caso de contar con el personal suficiente, no es totalmente necesario utilizar Alias, solo se tiene que colocar el nombre del responsable directo.

Fuente: Elaboración propia, 2017.

D. Diseño del prototipo de App Draw-Hánzí

A continuación, se describirán las tarjetas de historias, mismas cuyo boceto es ilustrado en la Tabla 45., se explicará brevemente cada apartado de las tarjetas, para la descripción de los requerimientos.

- **Historias de Usuarios**

- Número de requisito: en este apartado, se asignará un identificador, por si en lo sucesivo, se tiene que referirse a él en otros apartados posteriores.
- Tipo: en este apartado se colocará bajo que tipo de requerimiento se agrupará el mismo.

- Nombre de la historia: en este apartado se encuentra el nombre que se le asigno a la historia de usuario.
- Usuario: en este apartado se especifica quien inspira esta sección.
- Descripción: en este apartado, se describirá claramente que se quiere que la App cumpla exactamente.
- Justificación: en este apartado, se colocará cuál es el motivo exacto por el que se quiere cumplir este requerimiento.
- Criterio de Satisfacción: en este apartado se sabrá claramente si el requisito se ha cumplido adecuadamente.
- Satisfacción del cliente: en este apartado se valorará de 1 a 5 (de menos a más), cuan satisfecho se sentiría el cliente si se cumple el requerimiento.
- Insatisfacción del cliente: en este apartado se valorará de 1 a 5 (de menos a más), cuan insatisfecho se sentiría el cliente si no se cumple el requerimiento.
- Prioridad: en este apartado se valora el valor de importancia que tiene en la App que este requerimiento se cumpla.
 - Prioridad Alta: Requerimiento critico o necesario para el correcto funcionamiento del App.
 - Prioridad Media: Requerimiento excusable para el correcto funcionamiento del App.
 - Prioridad Baja: Requerimiento no necesario para el correcto funcionamiento del App.
- Puntos Estimados: en este apartado se genera un estado de equilibrio y ayuda a que el equipo se concentre, su calificación va desde 5 hasta 10 (de menos a más).
- Programador Responsable: en este apartado se identifica claramente a la persona encargada del desarrollo de la App.
- Observaciones: en este apartado se permite explicar el requerimiento más a específicamente, para evitar confusiones o aclarar dudas.

Tabla 45. Ejemplo de plantilla de Historia de Usuario

HISTORIA DE USUARIO			
Número de Requisito		Tipo	
Nombre de la Historia		Actor (usuario)	
Descripción			
Justificación			
Criterio de Satisfacción			
Satisfacción del Usuario		Insatisfacción Del Usuario	Prioridad
Puntos Estimados		Iteración Asignada	Programador Responsable
Observaciones			

Fuente: Elaboración propia, 2017.

Se procede a la redacción de las tarjetas de historias de usuario, obtenidas de las reuniones con el cliente y tratando de redactarlas de la forma más clara posible para poder identificar correctamente los requerimientos. En lo sucesivo se tratarán de ilustrar las más historias de usuario básicas que se podrían presentar en un proceso de desarrollo inicial, como lo muestran desde la Tabla 46 hasta la Tabla 61, respectivamente a manera de ejemplo, es decir, que no son las únicas historias que pueden surgir durante el proceso y tampoco son plenamente fijas, durante el mismo, ya que el cliente puede señalar un cambio, con base, en alguna de ellas.

Tabla 46. Historia de Usuario de “Diseño”

HISTORIA DE USUARIO			
Número de Requerimiento	#001	Tipo	Requisito de Apariencia
Nombre de la Historia	Diseño Atractivo	Actor	Usuario
Descripción	El diseño de la interfaz resulta muy atractivo para el usuario		
Justificación	La interfaz resultará agradable a la vista de los usuarios mediante una combinación de colores que deberá invitar a usuario a interaccionar fácilmente con nuestro sistema y de esta manera hacerlos sentir muy cómodos.		
Verificable	SI	Medio de Verificación	Mediante encuesta
Criterio de Satisfacción	Se realizará una breve encuesta a los usuarios que hayan interactuado con la App, más de una vez y se esperará que entre el 85% -90% o más confirme el atractivo del diseño final		
Satisfacción del Usuario	4	Insatisfacción Del Usuario	3
Puntos Estimados	8	Iteración Asignada	1
Observaciones	Utilizar preferentemente utilizar colores azules, negro, rojo, naranja, verde, amarillo, por mencionar algunos, en tonos vibrantes		
		Prioridad	Alta
		Programador Responsable	ProgGU ProgQS

Fuente: Elaboración propia, 2017.

Tabla 47. Historia de Usuario de “Estilo”

HISTORIA DE USUARIO					
Número de Requerimiento	#002	Tipo	Requisito de Estilo		
Nombre de la Historia	Diseño Sencillo	Actor	Usuario		
Descripción	El diseño de la interfaz deberá ser sencilla y ordenada				
Justificación	La App deberá ser simple y no se deberá de utilizar elementos que puedan ocasionar confusiones, ya que se tomara en cuenta que los usuarios finales no cuentan con un alto dominio de las Apps				
Verificable	SI	Medio de Verificación	Mediante encuesta		
Criterio de Satisfacción	Se realizará una breve encuesta a los usuarios que hayan interactuado con la App, más de una vez y se esperará que entre el 85% -90% o más confirme el atractivo del diseño final				
Satisfacción del Usuario	4	Insatisfacción Del Usuario	3	Prioridad	Alta
Puntos Estimados	9	Iteración Asignada	1	Programador Responsable	ProgGU ProgQS
Observaciones	Utilizar preferentemente más iconos que texto				

Fuente: Elaboración propia, 2017.

Tabla 48. Historia de Usuario de “Utilidad”

HISTORIA DE USUARIO					
Número de Requerimiento	#003	Tipo	Requisito de Utilidad		
Nombre de la Historia	Uso del App	Actor	Usuario		
Descripción	El App será muy fácil de utilizar por los usuarios				
Justificación	Como los usuarios son de diversas edades y estos mismos serán los mayores críticos para la misma, por ello el App será útil y simple en el momento de hacer uso de ella				
Verificable	SI	Medio de Verificación	Mediante testing de oportunidad		
Criterio de Satisfacción	Cuando se hagan las pruebas de testing con usuarios de oportunidad, se evaluará la interacción que estos tengan con la App y se irán midiendo los tiempos que dependiendo de sus características cada usuario necesite. El 98% de los Testers no debería tener problemas que impliquen un exceso de tiempo en las pruebas				
Satisfacción del Usuario	3	Insatisfacción Del Usuario	2	Prioridad	Alta
Puntos Estimados	6	Iteración Asignada	5	Programador Responsable	ProgGU ProgQS
Observaciones	El tiempo de respuesta ideal de la App es de 5 segundos				

Fuente: Elaboración propia, 2017.

Tabla 49. Historia de Usuario de “Experiencia”

HISTORIA DE USUARIO					
Número de Requerimiento	#004	Tipo	Requisito de Experiencia		
Nombre de la Historia	Conocimiento técnico	Actor	Usuario		
Descripción	El App se utiliza sin recibir una formación técnica previa				
Justificación	Los usuarios no deberán necesitar una formación para poder utilizar correctamente en el App				
Verificable	SI	Medio de Verificación	Mediante encuesta		
Criterio de Satisfacción	El 70% o más de las personas que no han interactuado con App similares y/o iguales, no deberán tener problemas con el uso de la App				
Satisfacción del Usuario	3	Insatisfacción Del Usuario	2	Prioridad	Alta
Puntos Estimados	8	Iteración Asignada	1	Programador Responsable	ProgGU ProgQS
Observaciones	No aplica				

Fuente: Elaboración propia, 2017.

Tabla 50. Historia de Usuario de “Atractivo”

HISTORIA DE USUARIO					
Número de Requerimiento	#005	Tipo	Requisito de Encanto		
Nombre de la Historia	Charm	Actor	Usuario		
Descripción	El App tendrá una notación lingüística semiformal cotidiano, lo que la hace atractiva y comprensiblemente sencilla para cualquier usuario, independientemente de su profesión y/o edad				
Justificación	Los usuarios de la App deben de entender el lenguaje de la misma sin presentar ningún tipo de complejidad				
Verificable	NO	Medio de Verificación	No aplica (observación)		
Criterio de Satisfacción	Debido a que el lenguaje es cotidiano y fácil de comprender, por lo cual no se deberán observar complicaciones y/o problema alguno con el lenguaje, en las pruebas de usabilidad, cuando los usuarios interactúan con la misma App				
Satisfacción del Usuario	5	Insatisfacción Del Usuario	4	Prioridad	Alta
Puntos Estimados	10	Iteración Asignada	1	Programador Responsable	ProgGU ProgQS
Observaciones	No aplica				

Fuente: Elaboración propia, 2017.

Tabla 51. Historia de Usuario de “Apariencia”

HISTORIA DE USUARIO					
Número de Requerimiento	#006	Tipo	Requisito de Universalidad		
Nombre de la Historia	Diseño visual	Actor	Usuario		
Descripción	El App deberá utilizar iconos universales y/o estándares, que son fáciles de identificar e interpretar				
Justificación	Con el uso de iconos ya conocidos, se evitará la confusión en el usuario y que el uso de la App sea considerada más intuitiva				
Verificable	NO	Medio de Verificación	No aplica (observación)		
Criterio de Satisfacción	Serán utilizados iconos prediseñados estandarizados				
Satisfacción del Usuario	5	Insatisfacción Del Usuario	4	Prioridad	Alta
Puntos Estimados	10	Iteración Asignada	1	Programador Responsable	ProgGU ProgQS
Observaciones	Se podrían utilizar iconos, que incluso hayan sido utilizados en otras Apps conocidas y que el usuario utiliza cotidianamente				

Fuente: Elaboración propia, 2017.

Tabla 52. Historia de Usuario de “Acceso”

HISTORIA DE USUARIO					
Número de Requerimiento	#007	Tipo	Requisito de Respuesta		
Nombre de la Historia	Tiempo de acceso	Actor	Usuario		
Descripción	La App funcionará con una demora máxima de 8 segundos en ejecutar cualquier acción				
Justificación	El sistema tiene que complacer al usuario en varios sentidos, Uno de ellos es en la rapidez de la respuesta para evitar que los usuarios dejen de usar el App				
Verificable	SI	Medio de Verificación	Mediante toma de tiempo y pequeña encuesta		
Criterio de Satisfacción	En la fase de pruebas, se medirá el tiempo exacto que puede llegar a tardar el proceso de carga de las diversas acciones de la App				
Satisfacción del Usuario	4	Insatisfacción Del Usuario	2	Prioridad	Alta
Puntos Estimados	6	Iteración Asignada	5	Programador Responsable	ProgGU ProgQS
Observaciones	No aplica				

Fuente: Elaboración propia, 2017.

Tabla 53. Historia de Usuario de “Legibilidad”

HISTORIA DE USUARIO					
Número de Requerimiento	#008	Tipo	Requisito de Visibilidad		
Nombre de la Historia	Legibilidad	Actor	Usuario		
Descripción	Los textos que se muestran en el App son completamente legibles				
Justificación	Los textos serán legibles para cualquier tipo de usuario, independientemente de la condición física de su vista				
Verificable	NO	Medio de Verificación	No Aplica (Observación)		
Criterio de Satisfacción	Un 90% o más de los usuarios deberán quedar satisfechas con la lectura de los textos.				
Satisfacción del Usuario	3	Insatisfacción Del Usuario	2	Prioridad	Alta
Puntos Estimados	6	Iteración Asignada	5	Programador Responsable	ProgGU ProgQS
Observaciones	Se deberán utilizar fuentes tipográficas conocidas para los usuarios				

Fuente: Elaboración propia, 2017.

Tabla 54. Historia de Usuario de “Disponibilidad”

HISTORIA DE USUARIO					
Número de Requerimiento	#009	Tipo	Requisito de Disponibilidad		
Nombre de la Historia	Asequibilidad	Actor	Usuario		
Descripción	El App deberá de estar disponible al menos el 99% del tiempo correctamente				
Justificación	El usuario deberá poder acceder al App en cualquier momento. Esto aumenta su fiabilidad y confianza en el servicio				
Verificable	NO	Medio de Verificación	No Aplica (Observación)		
Criterio de Satisfacción	Debido a que se usará Azure y este tiene un compromiso de nivel de servicios que ofrece una disponibilidad del 99% en todos sus usos				
Satisfacción del Usuario	5	Insatisfacción Del Usuario	4	Prioridad	Media
Puntos Estimados	8	Iteración Asignada	5	Programador Responsable	ProgGU ProgQS
Observaciones	No Aplica				

Fuente: Elaboración propia, 2017.

Tabla 55. Historia de Usuario de “Escalabilidad”

HISTORIA DE USUARIO					
Número de Requerimiento	#010	Tipo	Requisito de Escalabilidad		
Nombre de la Historia	Concurrencia	Actor	Usuario		
Descripción	La App deberá de ser escalable				
Justificación	Dependiendo del número de usuarios que deseen utilizar la App, la misma deberá poder soportar cualquier cambio petición del usuario en todo momento				
Verificable	NO	Medio de Verificación	No Aplica (Observación)		
Criterio de Satisfacción	Debido a que se usará Azure, a medida que aumente el conjunto de datos, la BD de Azure los distribuirá automáticamente entre los recursos existentes, para satisfacer las necesidades de almacenamiento correctamente				
Satisfacción del Usuario	5	Insatisfacción Del Usuario	2	Prioridad	Alta
Puntos Estimados	8	Iteración Asignada	5	Programador Responsable	ProgGU ProgQS
Observaciones	No Aplica				

Fuente: Elaboración propia, 2017.

Tabla 56. Historia de Usuario de “Lanzamiento”

HISTORIA DE USUARIO					
Número de Requerimiento	#011	Tipo	Requisito de Lanzamiento		
Nombre de la Historia	Publicación	Actor	Usuario		
Descripción	La App se irá actualizando sin ocasionar inconveniente alguno				
Justificación	La App se actualizará sin ocasionar molestia alguna a los usuarios y sin ocasionar fallo alguno a la misma				
Verificable	NO	Medio de Verificación	No Aplica (Observación)		
Criterio de Satisfacción	Dado que la App se ejecutará en la Cloud, se harán los cambios de manera Offline hasta colaborar el correcto funcionamiento y una vez aceptado se subirá la nueva versión al Cloud				
Satisfacción del Usuario	2	Insatisfacción Del Usuario	4	Prioridad	Alta
Puntos Estimados	5	Iteración Asignada	5	Programador Responsable	ProgGU ProgQS
Observaciones	No Aplica				

Fuente: Elaboración propia, 2017.

Tabla 57. Historia de Usuario de “Adaptabilidad”

HISTORIA DE USUARIO					
Número de Requerimiento	#012	Tipo	Requisito de Adaptabilidad		
Nombre de la Historia	Adaptación	Actor	Usuario		
Descripción	La App se podrá utilizar de manera totalmente correcta en los diferentes tamaños de las pantallas de los variados dispositivos existentes.				
Justificación	El usuario deberá poder utilizar la App desde el dispositivo específico que desee.				
Verificable	NO	Medio de Verificación	No Aplica (Observación)		
Criterio de Satisfacción	Se realizarán pruebas de usabilidad en los diferentes dispositivos, antes de su publicación y en todos los casos deben superarse satisfactoriamente				
Satisfacción del Usuario	3	Insatisfacción Del Usuario	5	Prioridad	Alta
Puntos Estimados	7	Iteración Asignada	1	Programador Responsable	ProgGU ProgQS
Observaciones	No Aplica				

Fuente: Elaboración propia, 2017.

Tabla 58. Historia de Usuario de “Entereza”

HISTORIA DE USUARIO					
Número de Requerimiento	#013	Tipo	Requisito de Integridad		
Nombre de la Historia	Integridad	Actor	Usuario		
Descripción	La App se podrá utilizar de manera totalmente correcta en el OS Android desde la versión 4.2 hasta la 7.0				
Justificación	El usuario debe poder utilizar la App desde el OS del dispositivo				
Verificable	NO	Medio de Verificación	No Aplica (Observación)		
Criterio de Satisfacción	Se realizarán pruebas de usabilidad en las diferentes versiones del OS Android, antes del lanzamiento de la App y en todos los casos deben superarse satisfactoriamente				
Satisfacción del Usuario	3	Insatisfacción Del Usuario	5	Prioridad	Alta
Puntos Estimados	7	Iteración Asignada	1	Programador Responsable	ProgGU ProgQS
Observaciones	No Aplica				

Fuente: Elaboración propia, 2017.

Tabla 59. Historia de Usuario de “Intromisión”

HISTORIA DE USUARIO					
Número de Requerimiento	#014	Tipo	Requisito de Seguridad		
Nombre de la Historia	Corrupción	Actor	Usuario		
Descripción	La App tendrá que definir restricciones que impidan corrupción en la integridad lógica de los datos, a la vez que no permite ningún tipo de error, para que el usuario no pierda ningún dato				
Justificación	La seguridad lógica de los datos es esencial para el éxito de la App. Deberá evitar que aparezcan datos en formatos incorrectos o corruptos.				
Verificable	NO	Medio de Verificación	No Aplica (Observación)		
Criterio de Satisfacción	Se tendrán que proporcionar desde etapas de desarrollo tempranas todas las facilidades para que los datos se almacenen correctamente y como el usuario lo desee				
Satisfacción del Usuario	2	Inatisfacción Del Usuario	3	Prioridad	Alta
Puntos Estimados	7	Iteración Asignada	1	Programador Responsable	ProgGU ProgQS
Observaciones	No Aplica				

Fuente: Elaboración propia, 2017.

Tabla 60. Historia de Usuario de “Inmunidad”

HISTORIA DE USUARIO					
Número de Requerimiento	#015	Tipo	Requisito de Inmunidad		
Nombre de la Historia	Protección	Actor	Usuario		
Descripción	La App deberá de estar protegido contra los ataques informáticos más habituales como: virus, spyware o código malicioso				
Justificación	Los ataques informáticos ponen en riesgo la disponibilidad de la App y de la privacidad de la misma, por lo que de no garantizarse la excelente seguridad el usuario podría perder la confianza en la misma				
Verificable	NO	Medio de Verificación	No Aplica (Observación)		
Criterio de Satisfacción	Se deberá de garantizar la seguridad del usuario, sobre cualquier ataque informático, animando a la robustez optima de la App				
Satisfacción del Usuario	2	Inatisfacción Del Usuario	5	Prioridad	Alta
Puntos Estimados	7	Iteración Asignada	1	Programador Responsable	ProgGU ProgQS
Observaciones	No Aplica				

Fuente: Elaboración propia, 2017.

- **Bocetos (Mockups) del prototipo de App Draw-Hánzí**

Se muestran a continuación los Mockups iniciales ilustrados en desde la Figuras 64 hasta la Figura 68, respectivamente.

Figura 64. Pantalla de Registro



Figura 65. Pantalla de Login



Figura 66. Pantalla de Menu principal



Autor: Elaboracion propia, 2017.

Autor: Elaboracion Propia, 2017.

Autor: Elaboracion propia, 2017.

Figura 67. Pantalla de Acciones



Figura 68. Pantalla de Lecciones y/o conocimiento



Autor: Elaboracion propia, 2017.

Autor: Elaboración Propia, 2017.

E. Planificación de fases

A continuación, se procede a elaborar una planificación general de las fases como lo ilustra la Tabla 61., a manera de facilitar la delegación de las mismas en el equipo de trabajo respectivamente, además de mostrar de forma fácil en que momento se llevaran a cabo las ceremonias correspondientes al desarrollo del prototipo de App Draw- Hanzí.

Tabla 61. Planificación de fases

FASE		
Iteración	Descripción	
EXPLORACION		
INICIALIZACION		
Iteración 0	<ul style="list-style-type: none"> • Establecimiento del proyecto • Análisis de requerimientos iniciales • Entrenamiento 	
PRODUCCION		
Iteración 1 Módulo de Traducción	<ul style="list-style-type: none"> • Implementación del módulo de traducción • Refinamiento y actualización de las historias de usuarios • Refinamiento de las interfaces 	<ul style="list-style-type: none"> • Lanzamiento #001 • Generación de pruebas • Generación de aceptación
Iteración 2 Módulo de Caligrafía	<ul style="list-style-type: none"> • Implementación del módulo de caligrafía • Refinamiento y actualización de las historias de usuarios • Refinamiento de las interfaces 	<ul style="list-style-type: none"> • Lanzamiento #002 • Generación de pruebas • Generación de aceptación
Iteración 3 Módulo de Reconocimiento de Patrones	<ul style="list-style-type: none"> • Implementación del módulo de reconocimiento de patrones • Refinamiento y actualización de las historias de usuarios • Refinamiento de las interfaces 	<ul style="list-style-type: none"> • Lanzamiento #003 • Generación de pruebas • Generación de aceptación
Iteración 4 Módulo de Usuarios	<ul style="list-style-type: none"> • Implementación del módulo de usuarios • Refinamiento y actualización de las historias de usuarios • Refinamiento de las interfaces 	<ul style="list-style-type: none"> • Lanzamiento #004 • Generación de pruebas • Generación de aceptación
Iteración 5 Módulo de Conocimiento	<ul style="list-style-type: none"> • Implementación del módulo de conocimiento • Refinamiento y actualización de las historias de usuarios • Refinamiento de las interfaces 	<ul style="list-style-type: none"> • Lanzamiento #005 • Generación de pruebas • Generación de aceptación
Iteración 6 Módulo de monitoreo	<ul style="list-style-type: none"> • Implementación del módulo de monitoreo • Refinamiento y actualización de las historias de usuarios • Refinamiento de las interfaces 	<ul style="list-style-type: none"> • Lanzamiento #006 • Generación de pruebas • Generación de aceptación
ESTABILIZACION		
Iteración 4 Módulo de Usuarios	<ul style="list-style-type: none"> • Refactorización del módulo de usuarios • Refinamiento de interfaces • Generación y ejecución de pruebas de 	<ul style="list-style-type: none"> • Lanzamiento #007
Iteración 5 Módulo de Conocimiento	<ul style="list-style-type: none"> • Refactorización del módulo de usuarios • Refinamiento de interfaces • Generación y ejecución de pruebas de aceptación 	<ul style="list-style-type: none"> • Lanzamiento #008
Iteración 6 Módulo de monitoreo	<ul style="list-style-type: none"> • Refactorización del módulo de usuarios • Refinamiento de interfaces • Generación y ejecución de pruebas de aceptación 	<ul style="list-style-type: none"> • Lanzamiento #009
PRUEBAS Y CORRECCION DE ERRORES		
Iteración 7 Módulo de pruebas y corrección de la App	<ul style="list-style-type: none"> • Se realizan las pruebas necesarias • Se evalúan las pruebas • Se realiza el análisis de resultados 	<ul style="list-style-type: none"> → Pruebas de aceptación final → Pruebas de tiempo de carga final → Pruebas de tiempo de acceso final

Fuente: Elaboración propia, 2017.

F. Lista de verificación de la fase de inicialización

Debido a que se culmina la planificación de la fase de exploración se decide, realizar el CheckList de control y medición, ilustrado en la Tabla 62., con resultados aprobatorios. De haber sido el caso contrario, no se recomienda avanzar en a la siguiente fase hasta no solucionarlo, correctamente.

Tabla 62. Checklist del plan de la fase de inicialización

LISTA DE VERIFICACIÓN DEL PLAN PROYECTO EN LA FASE DE INICIALIZACIÓN			
FASE 2. INICIALIZACIÓN			
Documentación	SI	NO	N/A
Los documentos a elaborar en el proyecto han sido incluidos en el plan del proyecto (p.ej. arquitectura y documentación de la interfaz de usuario)	X		
La duración de cada elemento ha sido incluida en el plan del proyecto (p.ej. contenido y actualización)	X		
Seguro de calidad	SI	NO	N/A
Los procedimientos de garantía de calidad se han definido en el plan del proyecto, para cada producto (p.ej. documentación, código y producto), incluidos actores y calendario.	X		

Fuente: Elaboración propia, 2017.

3.2.2.3 FASE DE PRODUCCIÓN

Se usa el desarrollo dirigido por pruebas (TDD), antes de iniciar el desarrollo de una funcionalidad debe existir una prueba que verifique su funcionamiento. En esta fase se lleva a cabo toda la implementación de los módulos.

El propósito en la fase de producción es implementar la funcionalidad requerida en el producto, mediante la aplicación iterativa y el ciclo de desarrollo incremental.

Los objetivos de la fase de producción son:

1. Implementar la funcionalidad del producto priorizando los requerimientos del cliente, hacia el producto.

2. Enfoque la funcionalidad del núcleo fundamental de su aplicación a principio del incremento para permitir múltiples ciclos de mejora.

Los criterios de entrada para la fase de producción son:

1. Se ha completado la fase de la iteración anterior.
2. Se han identificado los requisitos funcionales más importantes.
3. El equipo ha sido ensamblado y ha sido entrenado para el desarrollo.
4. Se ha establecido el entorno de desarrollo.

3.2.2.3.1. Iteración 1 Módulo de Traducción

Cada iteración tiene que proporcionar un resultado completo, un incremento del producto que sea potencialmente entregable, de manera que cuando el cliente, lo solicite sólo sea necesario un esfuerzo mínimo para que el producto esté disponible para ser utilizado.

Nota 10: Para fines prácticos de la presente investigación científica tecnológica, únicamente se mostrará la 1era., iteración a manera de ilustrar en lo general el proceso de implementación de la metodología Mobile-D™, en un proyecto de desarrollo.

Sin embargo, se deben de tener una serie de restricciones importantes que se consideraran para cada una de ellas:

1. En la reunión de planificación de la iteración, el equipo de desarrollo adquirió el compromiso de completar en la misma iteración unos requisitos concretos, baso su plan y organización en ellos. Cambiar los objetivos / requisitos de la iteración dificulta la concentración del equipo de desarrollo, basa la moral, teniendo como resultado una falta de compromiso.
2. El hecho de no poder cambiar los objetivos / requisitos de la iteración una vez iniciada, facilita que el cliente cumpla con su responsabilidad de conocer qué es lo más prioritario a desarrollar, antes de iniciar la iteración.
3. Los objetivos / requisitos que se están desarrollando eran los más prioritarios justo antes de in iniciar la iteración. Así, mismo debido a que las iteraciones son tan cortas como para que la probabilidad de cambios una vez iniciada la iteración sea mínima.

- **Modelado de los datos**

En este punto se representan las tablas de historias de usuario, por funcionalidad específica, para la primera iteración, como lo muestran respectivamente desde la Tabla 63 hasta la Tabla 68., a la vez se muestran desde la Figura 69 hasta la Figura 82., respectivamente.

Tabla 63. Historia de Usuario del reconocimiento de “Target’s”

Reconocer el marcador y presentar AR.							
Número/ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes de	Después de	Estimado	Gastado		
F01	NUEVO	Fácil	Fácil	25 Hrs		1	
		Moderado	Moderado				
		Difícil	Difícil				
Descripción							
La App deberá de poder reconocer el marcador, para este caso "un carácter chino" y presentar la traducción del mismo en AR, a manera de imagen estándar. Así mismo de su pronunciación nativa correcta.							
Fecha		Estado			Comentario		
27 / Marzo / 2017		Definido					

Fuente: elaboración propia, 2017.

Tabla 64. Historia de Usuario del reconocimiento de “Target’s-Objetos”

Reconocer el marcador y presentar AR.							
Número/ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes de	Después de	Estimado	Gastado		
F02	NUEVO	Fácil	Fácil	35 Hrs		2	
		Moderado	Moderado				
		Difícil	Difícil				
Descripción							
La App deberá de poder reconocer el marcador, para este caso "un objeto universal" y deberá presentar la traducción caligráfica en el carácter chino simplificado, así como su traducción pinyin.							
Fecha		Estado			Comentario		
03/ Abril / 2017		Definido					

Fuente: elaboración propia, 2017.

Tabla 65. Historia de Usuario del Reconocimiento de “Target’s-Video”

Reconocer el marcador y presentar AR.							
Número/ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes de	Después de	Estimado	Gastado		
F03	NUEVO	Fácil	Fácil	25 Hrs		1	
		Moderado	Moderado				
		Difícil	Difícil				
Descripción							
La App deberá de poder reconocer el marcador, para este caso “un carácter chino” y presentar el correcto sentido caligráfico de escritura del mismo.							
Fecha		Estado			Comentario		
10 / Abril / 2017		Definido					

Fuente: elaboración propia, 2017

Tabla 66. Tarjeta de tarea T01 relacionada con la función F01

Reconocer el marcador y presentar AR							
Número/ID	Tipo	Dificultad		Confianza	Esfuerzo		Notas
		Antes de	Después de	1 (pocas confianzas) 4 (muchas confianzas)	Estimado	Gastado	
F01-T01	NUEVO	Rutina (1)- Mucha Dificultad (5) 1		3	2 Hrs		
Descripción							
Creación de marcador (Tarjet) a identificar, con un carácter chino “especifico”, para poder ser identificado por la App							
Fecha		Estado			Comentario		
27 / Marzo / 2017		Definido					

Fuente: Elaboración propia, 2017.

Figura 69. Ejemplo de Target a identificar de “Ayuda”

The image shows the Chinese characters '帮助' (bāngzhù), which translates to 'help' or 'assistance'. The characters are rendered in a large, bold, black font with a slight shadow effect, set against a plain white background.

Fuente: Elaboración propia, 2017.

Tabla 67. Tarjeta de tarea T02 relacionada con la función F01

Reconocer el marcador y presentar AR							
Número/ID	Tipo	Dificultad		Confianza	Esfuerzo		Notas
		Antes de	Después de	1 (poca confianza) 4 (mucho confianza)	Estimado	Gastado	
F01-T02	NUEVO	Rutina (1)- Mucha Dificultad (5) 3		3	8 Hrs		
Descripción							
Creación de la base de datos, con el target's, así como con el target que se mostrara al detectarse el primero, en la plataforma de desarrollo VUFORIA.							
Fecha		Estado			Comentario		
27 / Marzo / 2017		Definido					

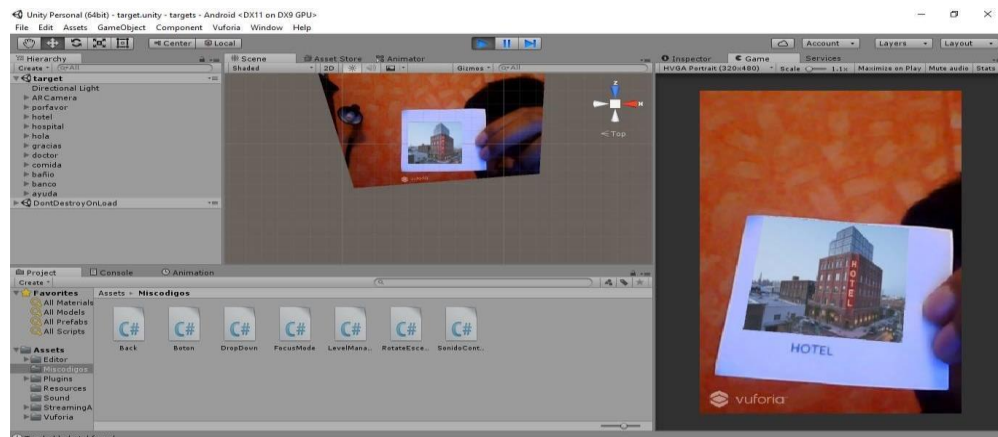
Fuente: elaboración propia, 2017.

Figura 70. Ejemplo de base de datos de los Target's en VUFORIA

target Name	type	Rating	Status	Date Modified
modelo	Single Image	★★★★★	Active	Jul 18, 2017 17:42
billete	Single Image	★★★★★	Active	Jul 18, 2017 17:41
porfavor	Single Image	★★★★★	Active	Mar 21, 2017 21:26
hotel	Single Image	★★★★★	Active	Mar 21, 2017 21:26
hospital	Single Image	★★★★★	Active	Mar 21, 2017 21:26
hola	Single Image	★★★★★	Active	Mar 21, 2017 21:26
gracias	Single Image	★★★★★	Active	Mar 21, 2017 21:26
doctor	Single Image	★★★★★	Active	Mar 21, 2017 21:25
comida	Single Image	★★★★★	Active	Mar 21, 2017 21:25
banio	Single Image	★★★★★	Active	Mar 21, 2017 21:25
banco	Single Image	★★★★★	Active	Mar 21, 2017 21:24
ayuda	Single Image	★★★★★	Active	Mar 21, 2017 21:24

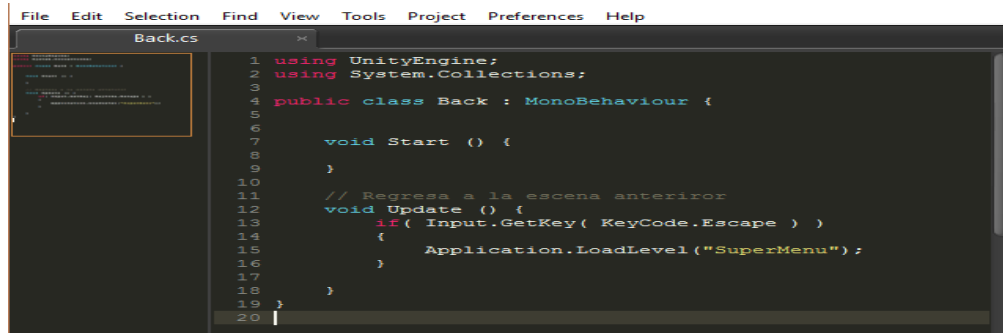
Fuente: Elaboración propia, 2017

Figura 71. Ejemplo de trabajo en el motor de desarrollo UNITY



Fuente: Elaboración propia, 2017.

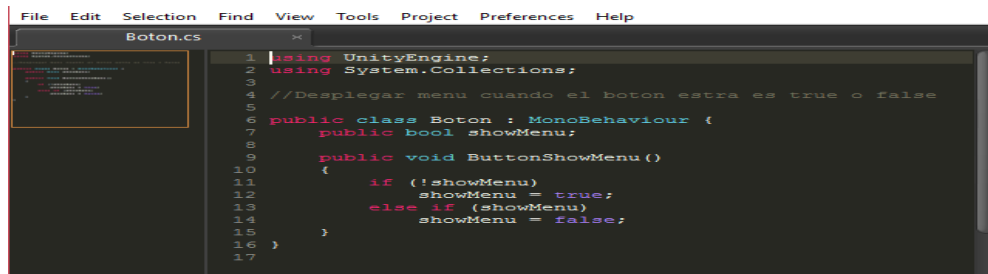
Figura 72. Ejemplo de código “Back” utilizado en el motor de desarrollo UNITY



```
File Edit Selection Find View Tools Project Preferences Help
Back.cs
1 using UnityEngine;
2 using System.Collections;
3
4 public class Back : MonoBehaviour {
5
6
7     void Start () {
8
9     }
10
11     // Regresa a la escena anterior
12     void Update () {
13         if ( Input.GetKey( KeyCode.Escape ) )
14         {
15             Application.LoadLevel( "SuperMenu" );
16         }
17     }
18 }
19
20
```

Fuente: Elaboración propia, 2017.

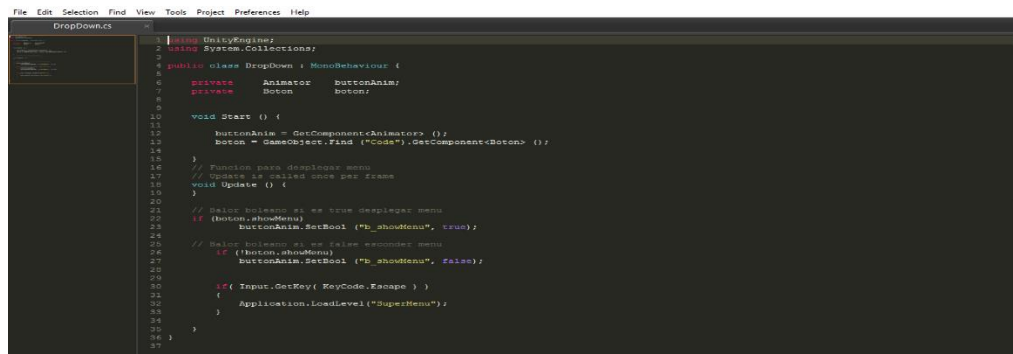
Figura 73. Ejemplo de código “Botón” utilizado en el motor de desarrollo UNITY



```
File Edit Selection Find View Tools Project Preferences Help
Boton.cs
1 using UnityEngine;
2 using System.Collections;
3
4 //Desplegar menu cuando el boton extra es true o false
5
6 public class Boton : MonoBehaviour {
7     public bool showMenu;
8
9     public void ButtonShowMenu()
10    {
11        if (!showMenu)
12            showMenu = true;
13        else if (showMenu)
14            showMenu = false;
15    }
16 }
17
```

Fuente: Elaboración propia, 2017.

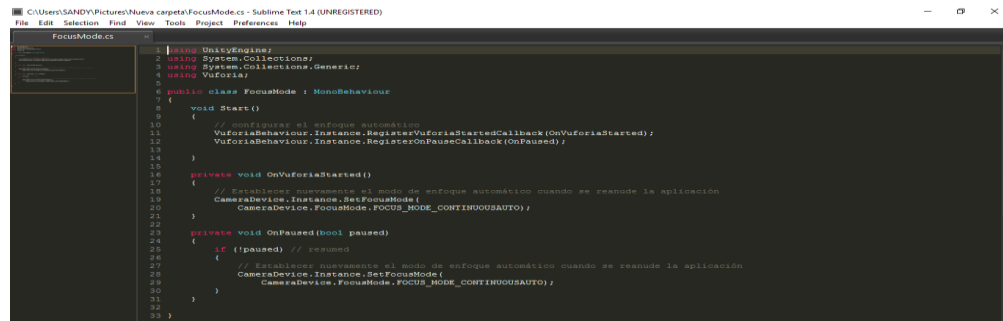
Figura 74. Ejemplo de código “DropDown” utilizado en el motor de desarrollo UNITY



```
File Edit Selection Find View Tools Project Preferences Help
DropDown.cs
1 using UnityEngine;
2 using System.Collections;
3
4 public class DropDown : MonoBehaviour {
5
6     private Animator buttonAnim;
7     private Boton boton;
8
9
10    void Start () {
11
12        buttonAnim = GetComponent<Animator> ();
13        boton = GameObject.Find ("Code").GetComponent<Boton> ();
14    }
15
16    // Funcion para desplegar menu
17    // Update is called once per frame
18    void Update () {
19
20    }
21
22    // Dale click al boton para desplegar menu
23    if (boton.showMenu)
24        buttonAnim.SetBool ("B_showMenu", true);
25
26    // Dale click al boton para desplegar menu
27    if (!boton.showMenu)
28        buttonAnim.SetBool ("B_showMenu", false);
29
30
31    if ( Input.GetKey( KeyCode.Escape ) )
32    {
33        Application.LoadLevel("SuperMenu");
34    }
35 }
36
37
```

Fuente: Elaboración propia, 2017.

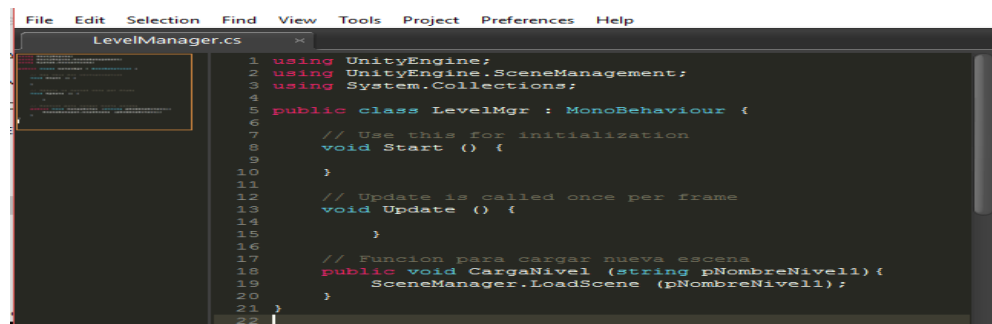
Figura 75. Ejemplo de código “FocusMode” utilizado en el motor de desarrollo UNITY



```
1 using UnityEngine;
2 using System.Collections;
3 using System.Collections.Generic;
4 using Vuforia;
5
6 public class FocusMode : MonoBehaviour
7 {
8     void Start()
9     {
10         // Configuración de enfoque automático
11         VuforiaBehaviour.Instance.RegisterVuforiaStartedCallback(OnVuforiaStarted);
12         VuforiaBehaviour.Instance.RegisterOnPauseCallback(OnPaused);
13     }
14
15     private void OnVuforiaStarted()
16     {
17         // Establecer nuevamente el modo de enfoque automático cuando se reanuda la aplicación
18         CameraDevice.Instance.SetFocusMode(
19             CameraDevice.FocusMode.FOCUS_MODE_CONTINUOUSAUTO);
20     }
21
22     private void OnPaused(bool paused)
23     {
24         if (!paused) // reanuda
25         {
26             // Establecer nuevamente el modo de enfoque automático cuando se reanuda la aplicación
27             CameraDevice.Instance.SetFocusMode(
28                 CameraDevice.FocusMode.FOCUS_MODE_CONTINUOUSAUTO);
29         }
30     }
31 }
32 }
```

Fuente: Elaboración propia, 2017.

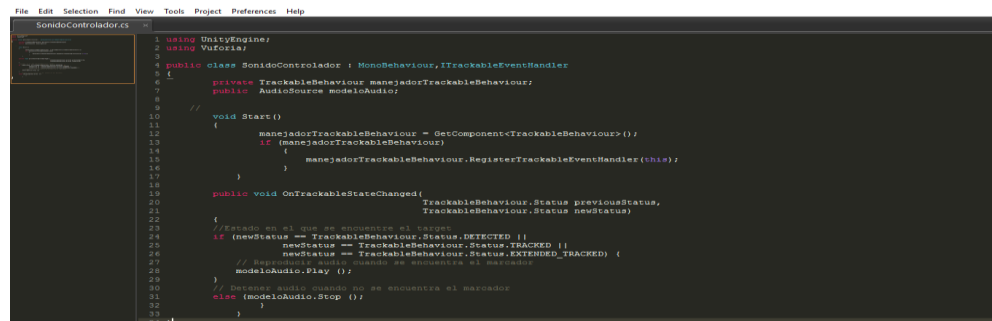
Figura 76. Ejemplo de código “LevelManager” utilizado en el motor de desarrollo UNITY



```
1 using UnityEngine;
2 using UnityEngine.SceneManagement;
3 using System.Collections;
4
5 public class LevelMgr : MonoBehaviour {
6
7     // Use this for initialization
8     void Start () {
9
10    }
11
12    // Update is called once per frame
13    void Update () {
14
15    }
16
17    // Funcion para cargar nueva escena
18    public void CargaNivel (string pNombreNivel){
19        SceneManager.LoadScene (pNombreNivel);
20    }
21 }
22 }
```

Fuente: Elaboración propia, 2017.

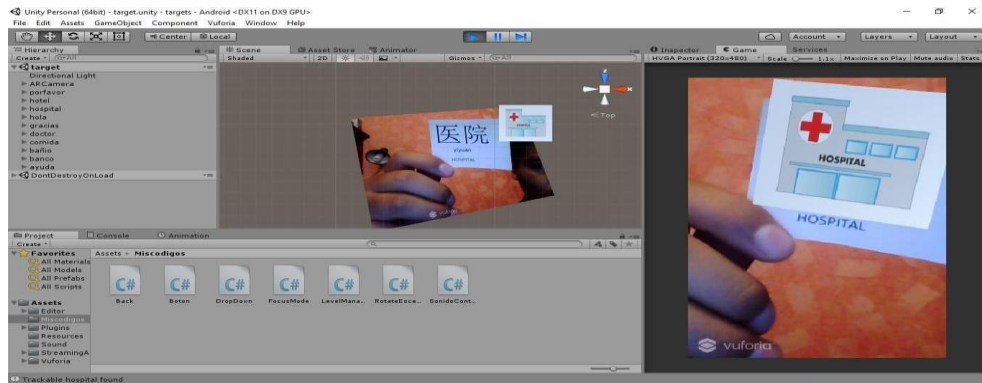
Figura 77. Ejemplo de código “SonidoControlador” utilizado en el motor de desarrollo UNITY



```
1 using UnityEngine;
2 using Vuforia;
3
4 public class SonidoControlador : MonoBehaviour, ITrackableEventHandler
5 {
6     private TrackableBehaviour manejadorTrackableBehaviour;
7     public AudioSource modeloAudio;
8
9     //
10    void Start()
11    {
12        manejadorTrackableBehaviour = GetComponent<TrackableBehaviour>();
13        if (manejadorTrackableBehaviour)
14        {
15            manejadorTrackableBehaviour.RegisterTrackableEventHandler(this);
16        }
17    }
18
19    public void OnTrackableStateChanged(
20        TrackableBehaviour.Status previousStatus,
21        TrackableBehaviour.Status newStatus)
22    {
23        //Estado en el que se encuentra el cargador
24        if (newStatus == TrackableBehaviour.Status.DETECTED ||
25            newStatus == TrackableBehaviour.Status.TRACKED ||
26            newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED) {
27            // Respuesta al estado en el que se encuentra el cargador
28            modeloAudio.Play ();
29        }
30        // Detiene audio cuando se encuentra el cargador
31        else {modeloAudio.Stop ();}
32    }
33 }
34 }
```

Fuente: Elaboración propia, 2017.

Figura 78. Ejemplo de simulación en el motor de desarrollo de UNITY.



Fuente: Elaboración propia, 2017.

Tabla 68. Tarjeta de tarea T01 relacionada con la función F03.

Reconocer el marcador y presentar AR								
Número/ID	Tipo	Dificultad		Confianza		Esfuerzo		Notas
		Antes de	Después de	1 (poca confianza)	4 (mucho confianza)	Estimado	Gastado	
F03- T01	NUEVO	Rutina (1)- Mucha Dificultad (5)	4	3		8 Hrs		
Descripción								
Generar el código para que se proyecte un video sobre la correcta "Caligrafía", al detectar una target específico.								
Fecha		Estado			Comentario			
15 / Abril / 2017		Definido						

Autor: Elaboración propia, 2017.

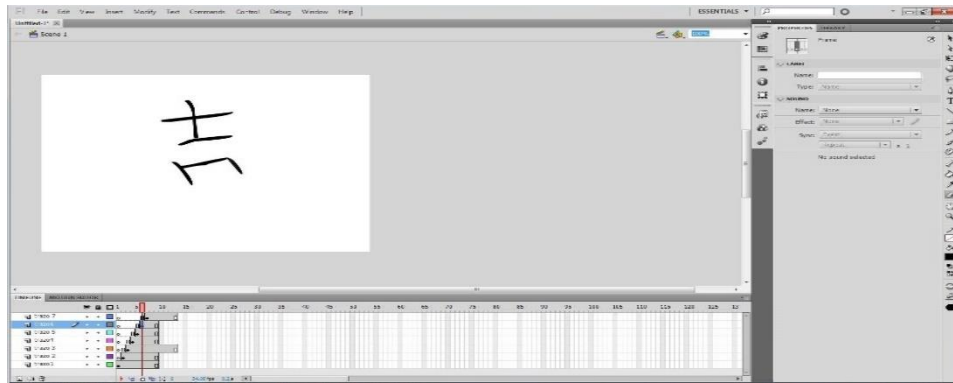
Figura 79. Ejemplo de código "Video Player" utilizado en el motor de desarrollo UNITY.

```

1 // Copyright (c) 2012-2014, Qualcomm Advanced Research Center, San Diego
2 // All Rights Reserved.
3 // Qualcomm Confidential and Proprietary. No disclosure, distribution, or
4 // reproduction is permitted without the express written permission of
5 // Qualcomm Incorporated. See http://www.qualcomm.com/legal/terms\_of\_use
6 // for more details.
7 // This code is provided as a reference only. It is not intended to be used
8 // as a substitute for the Qualcomm SDK, and is subject to all terms and
9 // conditions of the Qualcomm SDK License Agreement, which may be found at
10 // http://developer.qualcomm.com/legal/terms\_of\_use
11 // or http://www.qualcomm.com/legal/terms\_of\_use
12 //
13 // This code is provided as a reference only. It is not intended to be used
14 // as a substitute for the Qualcomm SDK, and is subject to all terms and
15 // conditions of the Qualcomm SDK License Agreement, which may be found at
16 // http://developer.qualcomm.com/legal/terms\_of\_use
17 // or http://www.qualcomm.com/legal/terms\_of\_use
18
19 #using UnityEngine
20 using UnityEngine.UI;
21
22 // This code is provided as a reference only. It is not intended to be used
23 // as a substitute for the Qualcomm SDK, and is subject to all terms and
24 // conditions of the Qualcomm SDK License Agreement, which may be found at
25 // http://developer.qualcomm.com/legal/terms\_of\_use
26 // or http://www.qualcomm.com/legal/terms\_of\_use
27
28 public class VideoPlaybackController : MonoBehaviour
29 {
30     #region PRIVATE_MEMBER_VARIABLES
31     private Vector2 mTouchStartPos;
32     private bool mTouchDown = false;
33     private float mTimeElapsed = 0.0f;
34     private bool mTapped = false;
35     private float mTimeToTap = 0.05f;
36     private bool mNonFullScreen = false;
37 }
    
```

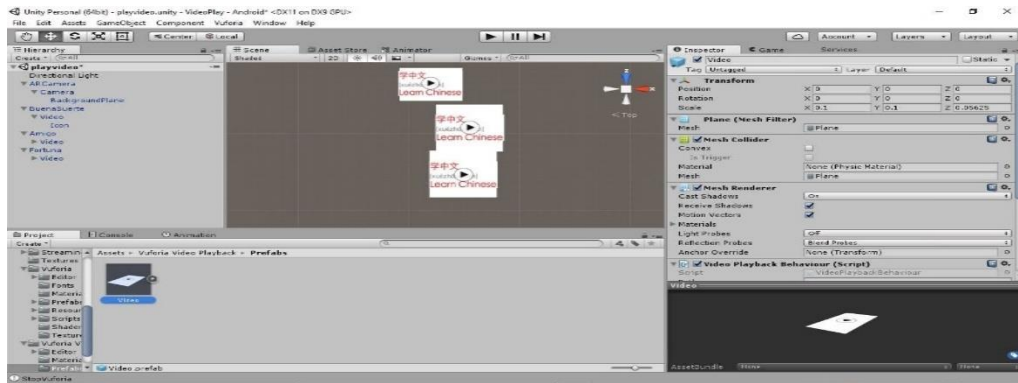
Fuente: QUALCOMM & Elaboración propia, 2017.

Figura 80. Ejemplo de creación de contenido para video en programa de diseño Flash.



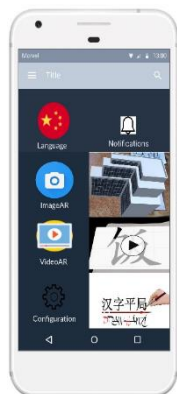
Fuente: Elaboración propia, 2017.

Figura 81. Ejemplo de simulación de video en el motor de desarrollo UNITY.



Fuente: Elaboración propia, 2017.

Figura 82. Ejemplo de prototipo preliminar de la App Draw- Hánzǐ.



Fuente: Elaboración propia, 2017.

3.2.2.4 FASE DE ESTABILIZACIÓN

El propósito de la fase de estabilización es asegurar la calidad de la implementación del proyecto.

Los objetivos de esta fase de estabilización son:

1. Finalizar la implementación funcional del producto de software del proyecto.
2. Mejorar y asegurar la calidad del producto.
3. Finalizar la documentación del producto.
4. La documentación final del producto.

Hay que tomar en cuenta que el propósito principal del presente proyecto de tesis, en el aspecto del desarrollo de la App Draw-Hánzǐ únicamente, tiene la intención de demostrar la facilidad de uso de la metodología ágil Mobile-DTM, para la creación de productos de software móvil, por lo que se define exclusivamente a la App Draw-Hánzǐ, ilustrada en las Figuras 83, 84, 85 donde se muestra la interfaz y el ejemplo de Target de reconocimiento, como un App prototipo, que servirá de referencia para modelos y diseños futuros, en la misma forma de producción. Así mismo la Figura 86, muestra las credenciales utilizadas como targets para el uso de los botones virtuales y acceso al examen on line.

No obstante, se logran estabilizar los siguientes aspectos del desarrollo (Véase Anexo IV, para observar su pleno funcionamiento):

Figura 83. Ejemplo de target final para reconocimiento.



Autor: Elaboración propia, 2017.

Figura 84. Ejemplo de fondo de pantalla final para la App Draw-Hánzǐ.



Autor: Freepik & Elaboración propia, 2017.

Figura 85. Ejemplo de fondo de pantalla final para la App Draw-Hánzǐ, en ejecución.



Autor: Elaboración propia, 2017.

Figura 86. Ejemplo de Credenciales de alumno.



Autor: Elaboración Propia, 2017.

3.2.2.5 FASE DE PRUEBAS Y CORRECCIÓN DE ERRORES

El propósito de la fase de pruebas y corrección de errores es ver si el sistema producido implementa correctamente la funcionalidad definida por el cliente, proporcionando la retroalimentación de la funcionalidad del sistema y corregir los defectos encontrados.

Los objetivos de la fase de pruebas y corrección de errores son:

1. Probar el sistema basado en la documentación producida en el proyecto.
2. Proporcionar información de defectos encontrados.
3. Dejar que el equipo desarrollador del proyecto planifique una solución, para los defectos encontrados.
4. Corregir los defectos.
5. Producir un sistema libre de errores, tanto como sea posible.

6. Lanzar un sistema con alta calidad, ya que las liberaciones han sido verificadas y validadas, así mismo las funcionalidades han sido implementadas y la documentación ha sido elaborada, lo que en conjunto aporta calidad al software.

Hay que tomar en cuenta que el propósito principal del presente proyecto de tesis e investigación científica tecnológica, era la producción prototipo de App Draw-Hánzí exclusivamente, con la intención de demostrar la facilidad de uso de la metodología ágil *Mobile-DTM*, en México, además al ser este el primer producto de software móvil que se fabrica y del que se toman las ideas más relevantes para la redacción del manual de desarrollo de Apps en México (Véase Anexo III), y de otros desarrollos consultados.

El prototipo de App Draw-Hánzí representa ya, la mayoría de las ideas en cuanto a diseño, soporte y tecnología esperados, además de cumplir el propósito con el que se formuló inicialmente por los autores de la presente investigación científica tecnológica inicialmente, así mismo, y conjuntamente a todo lo anterior, se le suma que el desarrollo, siempre fue pensado en lo particular (desarrollo vertical) y nunca se pensó en su comercialización masiva. Así mismo se muestran las historias de usuario y tareas terminadas en la fase de producción, tal como lo ilustra las Tablas. 69, 70, 71, respectivamente.

Tabla 69. Historia de Usuario #004 Requisitos de Experiencia, Terminada.

HISTORIA DE USUARIO				
Número de Requerimiento	#004	Tipo	Requisito de Experiencia	
Nombre de la Historia	Conocimiento técnico	Actor	Usuario	
Descripción	La App se podrá utilizar de manera totalmente correcta, sin ser estrictamente necesario recibir una formación técnica especializada previamente.			
Justificación	El usuario no deberá necesitar una formación específica necesaria para poder utilizar correctamente la App, independientemente de la edad, profesión y/o educación que este tenga. Únicamente deberá de ser necesario la experiencia que el usuario tenga en el uso diario de su dispositivo móvil.			
Verificable	SI	Medio de Verificación	Mediante pequeña encuesta oral, al usuario y/o observación práctica del mismo.	
Criterio de Satisfacción	Se realizarán pruebas de usabilidad en los diferentes dispositivos y con usuarios de oportunidad y/o testers, antes de su publicación final (versión de lanzamiento Beta) y en todos los casos deben superarse dicho requerimiento satisfactoriamente, de lo contrario, se regresará a la fase 4 de desarrollo por un periodo máximo de 2 semanas.			
Satisfacción del Usuario	10	Insatisfacción Del Usuario	2	Prioridad Alta
Puntos Estimados	8	Iteración Asignada	1	Programador Responsable ProgGU ProgQS
Observaciones	<ul style="list-style-type: none"> Influye considerablemente el rango de edad del usuario (manual de funcionamiento, ilustrado). Si es un factor decisivo el nivel de escolaridad en los rangos de edad menores a los estimados. Se observan un 50-50 de sugerencia de cambios de los iconos de la App. 			

Autor: Elaboración Propia, 2017.

Tabla 70. Historia de Usuario en funciones F01, Terminada.

RECONOCER EL MARCADOR Y PRESENTAR AR									
Número/ID	Tipo	Dificultad		Esfuerzo				Notas	
		Antes de	Después de	Estimado		Gastado			
F01	NUEVO	Fácil	Fácil	25	HRS.	12	HRS.	NO APLICA	
		Moderado	Moderado						
		Difícil	Difícil						
Descripción									
Generar el código correspondiente para que se la App pueda reconocer el Target especificado, procediendo a realizar la función de AR correspondiente (VIDEO, IMAGEN 2D y/o 3D, AUDIO, entre otras).									
Fecha	Estado		Comentario						
27 / Marzo / 2017	Iniciada		<ul style="list-style-type: none"> Se tiene que tomar muy en cuenta las especificaciones de cada uno de los gadgets, ya que afectan significativamente el funcionamiento. 						
28 / Marzo / 2017	En proceso de desarrollo		<ul style="list-style-type: none"> No existen muchos recursos gratuitos en 3D, hay que tomar en cuenta la posibilidad de desarrollar el contenido 3D. 						
29 / Marzo / 2017	En proceso de desarrollo		<ul style="list-style-type: none"> Se está presentando problema de estática con la nueva versión de UNITY. 						
30 / Marzo / 2017	Finalizada		<ul style="list-style-type: none"> Es vital el código de enfoque de la cámara del dispositivo (se sugiere documentarlo y anexarlo en el repositorio de código), para posteriores usos. 						

Autor: Elaboración propia, 2017.

Tabla 71. Tarjeta de Tarea F01-T02

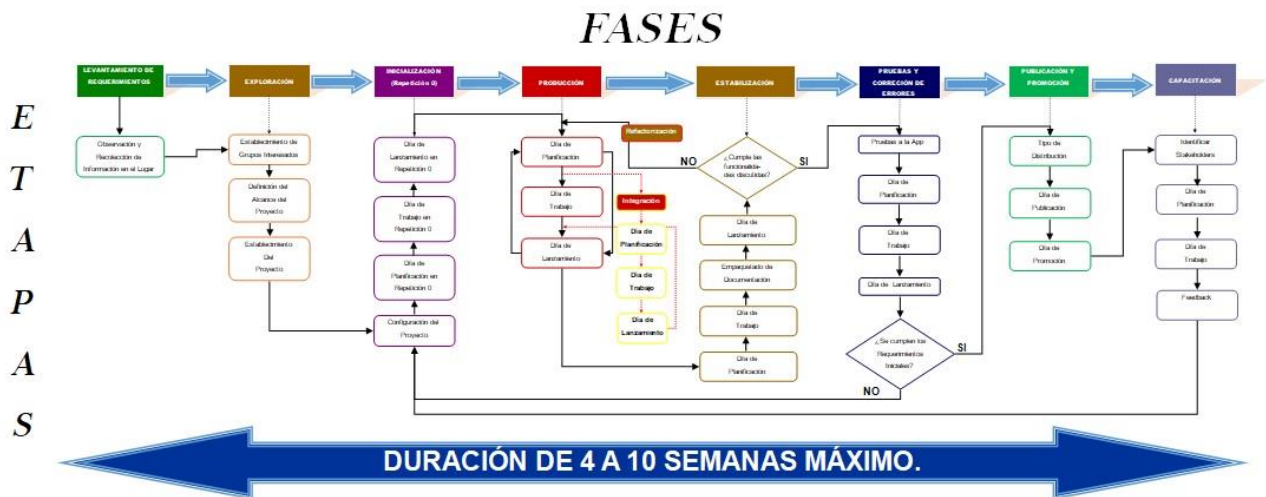
Tarjeta de tarea T02, relacionada con la función F01									
Número/ID	Tipo	Dificultad		Confianza	Esfuerzo				Notas
		Antes de	Después de	1 (poca confianza)	Estimado		Gastado		
		Rutina (1) - Mucha Dificultad (5)	Rutina (1) - Mucha Dificultad (5)	4 (mucho confianza)					
F01- T02	NUEVO	3	2	3	4	HRS.	2	HRS.	
Descripción									
Creación de la base de datos, con los diferentes target's, que se requerirán en el uso de presentación de la AR, en la herramienta de desarrollo VUFORIA y cargándola mediante el motor de desarrollo UNITY al proyecto de la App.									
Fecha	Estado		Comentario						
15 / Abril / 2017	Iniciada		<ul style="list-style-type: none"> Solo se pueden cargar 25 target's por base de datos. Es importante revisar que este actualizada la App Licence key, en el proyecto. 						
15 / Abril / 2017	Finalizada		<ul style="list-style-type: none"> Podemos realizar diferentes llamadas a múltiples BD, en la misma escena, dentro del proyecto (se recomienda documentar la acción e ingresarla al repositorio de operaciones). 						

Autor: Elaboración propia, 2017.

Con base, en lo anterior se ha decidido no lanzarlo en ninguna AppStore, no proseguir a su desarrollo completo y a dar por concluido el presente caso de estudio: Draw- Hánzí, de esta manera.

En consecuencia, de haber realizado el presente caso de estudio, con la App Draw-Hánzí, se determina que el diagrama de la Metodología Mobile-D™, queda finalmente de la siguiente manera, como lo ilustra la Figura 87, respectivamente.

Figura 87. Diagrama de la Metodología Mobile-D™, Final.



Autor: Elaboración Propia, 2017.

Capítulo IV

RESULTADOS Y CONCLUSIÓN

4. CONCLUSIÓN, RESULTADOS Y TRABAJOS FUTUROS

4.1 Resultados.

El desarrollo de un producto de software móvil “App”, sufre prácticamente de los mismos problemas que en su inmensa mayoría, sufre cualquier proceso de desarrollo de software, cualquiera que este sea, con o sin la experiencia que anteceda a su producción. Aunque si hay que recalcar las particularidades específicas del desarrollo de una App como son:

- 1) La corta duración de su producción.
- 2) La gran competencia del sector, lo que exige innovación y mucha actualización.
- 3) Los constantes cambios (novedades) en las plataformas de desarrollo (OS, lenguajes de programación, entre otros).
- 4) El infinito mundo de hardware en el que se puede ejecutar, así como sus variados y potentes beneficios.
- 5) La simplicidad y nuevo enfoque, que aportan algunas Apps revolucionarias, al modificar entornos sociales y/o modelos de negocios, ya existentes.
- 6) El alto número de rotación de personal en las empresas Start-up o PyME IT mexicanas y su alto costo de especialización.
- 7) La falta de tiempo, disposición y acceso a revisar las metodologías de desarrollo de software móvil a nivel mundial.

Todo lo anterior influye sobremanera, al momento de elegir la metodología de desarrollo de software móvil más correcta y óptima a las necesidades de producción de los desarrolladores de cualquier estrato (Freelance, Start-up y/o PyME IT), en México.

Actualmente en México, este mercado de desarrollo es considerado relativamente nuevo por lo que no se cuenta, con una metodología de producción, establecida, además de que no se cuenta con toda la infraestructura necesaria o que el mercado demanda, salvo contadas excepciones, la mayor parte de los proyectos de desarrollo de software móvil, en México, se llevan a cabo por equipos desarrolladores relativamente muy pequeños, que requieren un método de desarrollo más práctico de ejecutar, que permita organizar las tareas y actividades, de una forma fácil y sencilla, al mismo

tiempo de que no les implique demasiado costoso (tiempo-esfuerzo), de manera que el primer resultado del presente trabajo de tesis es la creación del “Manual de desarrollo de Apps en México”, que se muestra en la Figura 88., y puede ser consultado en su totalidad en el Anexo III.

Figura 88. Ejemplo de módulos del manual de desarrollo de Apps en México, sustentado en la metodología ágil *Mobile-DTM*



Fuente: Elaboración propia, 2017.

Consecuentemente a este resultado, se han obtenido también los siguientes:

1. Se utilizaron los conocimientos de ingeniería de Software e ingeniería de requerimientos adquiridos durante el transcurso de la Licenciatura en Informática Administrativa y algunos adquiridos de forma autodidacta, para la correcta interpretación de la metodología ágil Mobile-D™.
2. Se realizó un prototipo de App: Draw-Hánzǐ funcional para la enseñanza del idioma chino, satisfactoriamente de acuerdo a los requerimientos del cliente, como lo ilustra la Figura 89., se puede consultar la ejecución total de la App Draw-Hánzǐ en el Anexo IV.

Figura 89. Ejemplo de ejecución de la App Draw-Hánzǐ



Fuente: Elaboración propia, 2017.

3. Se logró la especialización de los autores del presente proyecto de tesis en la manipulación y generación de AR, en dispositivos móviles Android, como lo ilustra la Figura 90.

Figura 90. Ejemplo de Trabajo en motor de desarrollo UNITY.



Fuente: Elaboración propia, 2017.

4. Se dio certeza en el correcto proceso de desarrollo de una App, demostrada en el prototipo de App Draw- Hází. Aportando una oportuna administración de tareas y actividades de dicho proceso de desarrollo.
5. Se ha demostrado existe una causación, con una correlación positiva fuerte, entre el desconocimiento de una metodología de desarrollo de Apps en México y el desuso de las mismas; en la producción de productos de software móvil, lo que genera una mala experiencia e insatisfacción de los clientes, usuarios y/o consumidores de estos productos y que solo tienen acceso por el momento en su mayoría Apps de baja calidad.

“Se demostró que con el correcto uso de la metodología ágil Mobile-D™, en la creación y producción de una App, cualquiera, en un equipo de desarrollo pequeño y con mínima experiencia en el desarrollo de productos de software en México, es lo más adecuado, debido a que genera una mayor comprensión. Así mismo se pueden llevar a cabo simultáneamente varios procesos de producción del producto a desarrollar lo que crea una mayor productividad intrínseca. Debido a que es un modelo iteracional, simplifica el proceso de entrega vs. validación, lo que permite adoptar fácilmente cambios sobre la marcha y llevarlos a éxito con mayor practicidad. Debido a las justas entregas (artefactos), funcionales del producto, ayudan a no presentar una desviación de la entrega final contra la definición inicial del producto, a la vez las entregas permiten enfocar el talento a las zonas y/o funciones de mayor riesgo. Debido a que el cliente está en contacto constante con el equipo de desarrollo y recibe oportunamente las entregas funcionales, le genera una satisfacción importante por parte del mismo, ya que le permite observar los posibles cambios para poder satisfacer plenamente sus necesidades propias, de su marca, de sus procesos y/o comerciales”.

4.2 Trabajo Futuro.

- **En relación con la App.**

Como trabajo futuro entorno a la App Draw- Hází, se pretendió a largo plazo, aumentar las funcionalidades y experiencia de la misma, introduciendo el uso de la virtualidad aumentada (AV, por sus siglas en inglés), para poder utilizar tecnología de hardware, tal como: HoloLens, Control por voz, Geolocalización, por mencionar algunas. En un entorno virtual aumentado al 75%, lo que permita inducir emociones, sumada a la experiencia “vivencia”, lo que se traduce para el usuario como un objetivo de experimentar emociones específicas de conocimiento- viaje- aprendizaje- dentro de un espacio virtual, lo que le permita evaluar las potencialidades de la AV. Además, de aumentar los OS móviles, en que se pueda ejecutar la App Draw- Hází como son: iOS y Windows Phone, algo muy parecido a lo que se ilustra en la Figura 91.

Figura 91. Ejemplo de trabajo futuro.



Fuente: Information Age, 2017.

- **En relación con la metodología ágil Mobile-D™ en México.**

Como trabajo futuro entorno a la metodología ágil *Mobile-D™*, se pretendió contribuir a la traducción al español (México) de los diversos artículos, trabajos y documentos, encontrados dentro de la presente investigación científica tecnológica de tesis y que no se incluyeron completamente en el mismo, por falta de tiempo y espacio. Así mismo, se pretendió cofundar la comunidad *Mobile-D™* México, por medio de una página web y en las diversas redes sociales, como lo ilustra la Figura 92., a manera de

generar un foro de apoyo entre los usuarios de la misma metodología ágil: a largo plazo, con la experiencia, además del conocimiento, desbordar en esta el aprendizaje obtenido, que ayude a implementarla correctamente, en las PyMEs IT, Start-ups y/o desarrolladores Freelance en México, con resultados satisfactorios y de éxito.

Figura 92. Página de Facebook de la Comunidad Methodology Mobile-D™ en México



Fuente: Elaboración propia, 2017.

4.3 Conclusión.

En los diversos lugares y momentos, en los que se ha tenido acceso, por parte de los autores, de poder escuchar e interactuar, con las experiencias de los clientes en el sector de las Start-ups y las PyMEs Tecnológicas, así como del sector de desarrollo Freelance en México, denota lo crucial de su importancia, sobre la exigencia urgente de la necesidad de la “alta calidad” en el desarrollo de Apps en estos sectores, en algunos casos es tal el reclamo, que los mismos cliente, usuarios y/o consumidores, sugieren poner en marcha un estándar de metodología de requisitos (de preferencia internacional), conjuntamente con una estrategia de diseño que facilite la construcción de las mismas “Apps”, haciéndolas cada vez más eficientes, funcionales y seguras a sus necesidades.

Debido a estas múltiples experiencias desastrosas, que los clientes han experimentado al tratar de desarrollar (con la ayuda de terceros) software móvil, tanto en cuestión de tiempo, económicamente y de desgaste organizativo. Es que se torna de vital importancia que el cliente, pueda tener la sensación (tangible), de desarrollar el producto, durante el proceso mismo, lo que genera satisfacción intrínseca, situación que aporta con facilidad la metodología *Mobile-DTM*

Se comprueba que la decisión de utilizar la metodología *Mobile-DTM*, ha sido la correcta; debido a los múltiples beneficios que aportó al desarrollo del prototipo de App “Draw-Hánzí”, desde el punto de vista cliente al lograr que el mismo este en mayor contacto con el equipo desarrollador, lo que garantiza una mayor definición y la correcta priorización su idea (requerimientos), volviendo el desarrollo más productivo, para ambas partes y garantizando la calidad implícita al tener la seguridad de que se comprendió, se agotó y se optimizó, hasta el último recurso disponible, ofreciendo el producto final en el tiempo y forma precisa.

4.3.1 Aportes Personales de los autores

Se considera por parte de los autores del presente proyecto de tesis, que el mismo otorga los siguientes aportes:

- » La comprensión del paradigma, sobre la complejidad del desarrollo de Apps en el mundo, lo que permitió acceder a la dinámica práctica sobre las relaciones y la constitución, que todo organismo desarrollador de software posee; el ejercicio práctico de dicha idea de App proveyó de experiencia práctica a los autores y/o lectores del mismo.
- » El conocimiento de la metodología *Mobile-DTM* y sus técnicas de desarrollo ayudo, al acceso de la construcción de nuevos y comprensión de los no tan nuevos significados, en el entorno práctico del desarrollo de Apps mundialmente.
- » Permite la creación de una cultura mexicana, adepta a la disciplina agilista en el desarrollo de Apps con el propósito de satisfacer la calidad, esperada por el cliente de forma eficaz.
- » Abre la puerta para que se continúe el ejercicio de una investigación científica tecnológica, sobre las diversas metodologías ágiles y teorías, para el desarrollo de

Apps en el mundo, con los antecedentes que cita este presente proyecto. Proveyendo un espacio para la profundización de las investigaciones de este tipo en México.

- » Ayudo a construir un canal de información relevante sobre la metodología *Mobile-DTM*, permitiendo estar un poco más al pendiente de los pertinentes cambios en este sector a nivel mundial.

“A nivel personal, siempre me inquieto como surgían las ideas que persistentemente veía plasmadas en Apps, y que utilizo en mi día a día. El poder vivenciar el proceso de desarrollo de una App, con el uso de una metodología ágil como Mobile-DTM, me ha mostrado más claramente las áreas en las que tengo mayor habilidad, mismas que se podrían vislumbrar como mis áreas de oportunidad en el desarrollo profesional. Además de que me ha expuesto un mundo de especialización, hasta el punto inicial de la presente investigación, un tanto desconocido y para nada inspeccionado por mi parte, y que además de haber representado un gran reto a nivel de investigación, debido a que alguna parte de la información no era accesible con facilidad, también simboliza un área de mayor conocimiento que me gustaría adquirir a la brevedad.

Desde el punto de vista profesional el utilizar la metodología Mobile-DTM, me ha marcado un sin número de puntos importantes (pero que la mayoría de las ocasiones, se muestran insignificantes), que se deben de tomar en cuenta en todo desarrollo tecnológico, cualquiera que este sea y que de no ser por este acercamiento practico, considero hubiera sido imposible ver a tiempo y hubiera generado múltiples errores de novato. No obstante, errores que implican un costo: en tiempo, esfuerzo, dinero y conocimiento, lo que a la vez se hubiera traducido en una frustración innecesaria.

El aporte significativo del presente proyecto de tesis de investigación, en mi vida ha sido brutalmente importante desde cualquier punto en el que se quiera inspeccionar y aunque el camino fue bastante largo y tortuoso, ha dejado una huella inmensa de experiencia y conocimiento, lo que me genera una enorme satisfacción, aporta una linda sensación de logro, que forja la energía necesaria para impulsar nuevos retos y nuevos objetivos, de evolución personal y profesional, en conclusión es sorprendente como voy adquiriendo conocimiento sobre un tema y cada vez sé más poco sobre el mismo; las cosas no se van tornando más comprensibles, sino por el contrario cada día son más misteriosas y complejas. No obstante, es asombroso lo mucho que conozco del tema en la actualidad, pero doblemente asombroso todavía, es que, ese pequeñísimo escalón en mi formación del conocimiento, pueda conceder tanta sensación de poder y éxito”

Sandy Trejo Quintero, 2017.

“Al principio fue un poco complicado trabajar con una metodología ágil y mucho más con una tan poco conocida como es el caso de Mobile-DTM, debido que en lo personal nunca había trabajado con una. Pero gracias a este trabajo de investigación, he aprendido que usando una metodología como esta se puede simplificar el duro trabajo que conlleva el proceso de desarrollo de una app como Draw-HánzÍ, además de que me ayudo a adquirir nuevos conocimientos que me permitirán crecer y desempeñarme de manera más eficaz en el mercado laboral.

En lo personal pienso que las metodologías ágiles deben enfocarse a la simplicidad, ya que esta puede facilitar el trabajo de todos los miembros del equipo. En base a mi experiencia, he conocido profesionales que trabajan con una metodología ágil y otras que no por lo que me ha hecho, cuestionar que no es lo mismo ser una persona eficiente, que ser una persona que se adapta a trabajar con una metodología ágil, debido a que las personas eficientes no trabajan con una metodología por lo que tienden a cometer errores y por ende ponen en riesgo el trabajo de todo el equipo, caso contrario de los que trabajan con una, ya que estas cometen menos errores reduciendo riesgos y ofreciendo un producto de mayor calidad.

Por lo que el aporte de este proyecto de investigación en mi vida ha sido muy positivo, ya que me permitió obtener conocimientos que pude aplicar además de desarrollar nuevas habilidades, aunque el tiempo de elaboración de esta investigación fue bastante largo, puedo decir que ha cambiado mi forma de pensar debido a que antes pensaba que el desarrollo de apps era bastante compleja y tedioso, en conclusión es impresionante lo mucho que hay que saber sobre el tema y que cada vez surgen nuevas metodologías conforme la tecnología avanza por lo que es nuestro deber actualizarnos para poder hacer cosas más impresionantes y complejas.”

Ulises Moreno Gorgonio, 2017.

Bibliografía

- » Aipoly Inc. . (16 de 03 de 2017). *Google Play & iTunes Preview*. Obtenido de Aipoly Vision : <https://play.google.com/store/apps/details?id=com.aipoly.vision&hl=es>.
- » Álvarez, F. J., Hurtado Alegría, J. A., Mondragón , A. M., Muñoz Arteaga, J., Velázquez Amador, C. E., & Hernández Bieliukas, Y. C. (2014). *Gestion de Poryectos de Software*. ALFA III Europe Aid.
- » *Android Studio Blog Developer*. (s.f.). Obtenido de Android Studio: <https://developer.android.com/studio/index.html?hl=es-419#resources>.
- » AnkiDroid Open Source Team, & Admium Corp. . (28 de 02 de 2017). *Google Play & iTunes Preview*. Obtenido de Trajetas AnkiDroid & AnkiApp Flashcards: https://play.google.com/store/apps/details?id=com.ichi2.anki&hl=es_419
<https://itunes.apple.com/es/app/ankiapp-flashcards/id689185915?mt=8>.
- » Appio. (June de 2013). *Tipos de Apps: Nativas. Híbridas y Web Apps*. Obtenido de <http://appio.es/tipos-de-apps/>.
- » BlackBerry. (2017). *BlackBerry*. Obtenido de BlackBerry 10 OS: <https://global.blackberry.com/es/software/smartphones/blackberry-10-os>.
- » Blanco, P., Camarero, J., Fumero, A., Weterski, A., & Rodríguez , P. (2009). *Metodología de Desarrollo Ágil para Sistemas Móviles Introducción al desarrollo con Android y el iPhone*. España: Universidad Politecnica de Madrid. Obtenido de http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf.
- » CamFind Inc. (02 de 03 de 2016). *Google Play & iTunes Preview*. Obtenido de TapTapSee:
<https://play.google.com/store/apps/details?id=com.msearcher.taptapsee.android&hl=es>
<https://itunes.apple.com/us/app/taptapsee-blind-visually-impaired-camera/id567635020?mt=8>.

- » Cantone, D. (2008). *Biblia del programador implementación y debugging: claves, tecnicas y herramientas para construir codigo solido y confiable*. Buenos Aires, Argentina: MP Ediciones. Obtenido de <https://.wikispaces.com/file/view/ciclosdevidadelsoftware.pdf/579330701/ciclosdevidadelsoftware.pdf>.

- » Chan, K. (22 de 12 de 2016). World Economic Forum. *¿Qué idioma te permite interactuar de manera integral con la humanidad?* Obtenido de <https://www.weforum.org/es/agenda/2016/12/que-idioma-te-permite-interactuar-de-manera-mas-integral-con-la-humanidad/>.

- » Chicaiza Angamarca, D., & Morales Caiza, D. (2016). Implementación de un Sistema Domótico de Videovigilancia, aplicando la Metodología Mobile-D, sobre la Plataforma Android para el Control de la Seguridad de las Salas de Docentes de la Unidad Académica C.I.Y.A- La matriz universidad Técnica de Cotopaxi. *Implementación de un Sistema Domótico de Videovigilancia, aplicando la Metodología Mobile-D, sobre la Plataforma Android para el Control de la Seguridad de las Salas de Docentes de la Unidad Académica C.I.Y.A- La matriz universidad Técnica de Cotopaxi*. Latacunga, Ecuador.

- » ChinesePod Limited. (31 de 10 de 2016). *Google Play*. Obtenido de ChinesePod Next: <https://play.google.com/store/apps/details?id=com.chinesepod.next&hl=es>.

- » ChineseSkill Co., Ltd. . (16 de 05 de 2017). *Google Play & iTunes Preview*. Obtenido de Chinese Skill: <http://www.chinese-skill.com/cs.html>.

- » Chrissis, M. B., Konrad, M., & Shrum, S. (2009). CMMI Guía para la Integración de Procesos y la Mejora de Productos. En M. B. Chrissis, M. Konrad, & S. Shrum, *CMMI Guía para la Integración de Procesos y la Mejora de Productos*. (Segunda Edición ed., pág. 630). Pearson Educación S.A. Obtenido de <http://www.sei.cmu.edu/library/assets/cmmi-dev-v12-spanish.pdf>.

- » Cruz, A. (14 de Enero de 2014). Desarrollo Libre Desarrollo Web, Android y mucho más. *Realidad Aumentada Vuforia*. Obtenido de <http://www.desarrollolibre.net/blog/tema/73/android/realidad-aumentada-con-vuforia>.

- » Domenech, E. Á., & Abad, A. M. (12 de Marzo de 2013). Comparative Evaluation of Delfdroid whit XP and Scrum using the 4-DAT. *Revista Cubana de Ciencias Infromáticas*, 7. Obtenido de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992013000100003.

- » Dualingo . (2017). *Dualingo*. Obtenido de Dualingo: <https://es.duolingo.com/>.

- » Dueñas, E. (9 de Septiembre de 2015). Dispositivos Móviles. *Tec21 en tu escuela*. México: Tecnológico de Monterrey. Obtenido de <http://conoce.ver.itesm.mx/portfolio-item/dispositivos-moviles/>.

- » EcuRed Conocimiento con todos y para todos. (02 de 05 de 2017). *EcuRed Conocimiento con todos y para todos*. Obtenido de ISO/IEC 25000: https://www.ecured.cu/ISO/IEC_25000.

- » Escobar Bonilla, G. S., & Campaña Guzmán, A. I. (Mayo de 2014). Diseño e Implementación de una Aplicación Móvil que cumpla la función de estación en tierra para el monitoreo de UAV´S en el centro de investigación y desarrollo de la fuerza aérea ecuatoriana. *Diseño e Implementación de una Aplicación Móvil que cumpla la función de estación en tierra para el monitoreo de UAV´S en el centro de investigación y desarrollo de la fuerza aérea ecuatoriana*. Latacunga, Ecuador. Obtenido de <http://repositorio.espe.edu.ec/bitstream/21000/8188/1/AC-ESPEL-SOF-0002.pdf>.

- » Florencia Barrera, Á. F., Martínez Sanchez , V. M., & Ramírez Domínguez, L. M. (3 de Febrero de 2015). Sistema para dispositivos móviles usando cómputo ubicuo para servicios en la Escuela Superior de Cómputo. México: Instituto Politécnico Nacional ESCOM.

- » FluentFlix Limited. (01 de Mayo de 2017). *iTunes Preview*. Obtenido de FluentU- Aprender un idioma con videos!: <https://itunes.apple.com/us/app/fluentu-learn-a-language-with-videos/id917892175?mt=8>.
- » Fowler, M., & Highsmith, J. (Marzo de 2001). *www.agileAlliance.org*. Obtenido de www.agileAlliance.org: <http://agilemanifesto.org/iso/es/manifesto.html>.
- » Fuzi, C. (22 de Julio de 2013). Metodología Ágil para el Diseño de Aplicaciones Multimedia de Dispositivos Móviles Investigación Documental. *Metodología Ágil para el Diseño de Aplicaciones Multimedia de Dispositivos Móviles Investigación Documental*. Obtenido de <https://es.slideshare.net/cristopherf/metodologia-agil-para-el-diseo-de-aplicaciones-multimedias-moviles>.
- » Galan Sempere, E. M. (12 de 05 de 2013). *Alquibla, una mirada al mundo de las bibliotecas*. Obtenido de La escritura China: <http://www.alquiblaweb.com/2013/05/12/la-escritura-china/>.
- » Garzás, J. (12 de 01 de 2015). *javiergarzas.com*. Obtenido de Que no te confundan, lo que realmente son Daily meetings, o Reuniones Diarias, efectivos.: <http://www.javiergarzas.com/2015/01/daily-meetings-o-reuniones-diarias-efectivos.html>.
- » GCF Aprende Libre. (1998-2016). *GCF Aprende Libre*. Obtenido de Sistemas Operativos para Dispositivos Mviles: https://www.gcfaprendelibre.org/tecnologia/curso/informatica_basica/sistemas_operativos/5.do.
- » GestioPolis.com, E. (02 de Abril de 2001). ¿Qué es Seis Sigma? Metodología e implementación. *¿Qué es Seis Sigma? Metodología e implementación*. Obtenido de <https://www.gestiopolis.com/que-es-seis-sigma-metodologia-e-implementacion/>

- » Gilibets, L. (31 de Julio de 2013). *Comunidad IEBS*. Obtenido de Qué es la metodología Kanban y cómo utilizarla: <http://comunidad.iebschool.com/iebs/general/metodologia-kanban/>.
- » Gómez, J. (19 de 11 de 2013). *Yuma´s World Tecnología y algo más*. Obtenido de ¿Cómo es un Sprint Planning en Scrum?: <https://juanmagomez.wordpress.com/2013/11/19/como-es-un-sprint-planning-en-scrum/>.
- » González Morcillo, C., Vallejo Fernández , D., Albusac Jiménez, J. A., & Castro Sánchez , J. J. (2012). *Realidad Aumentada Un Enfoque Práctico con ARToolKit y Blender*. Ciudad Real, España: Bubok Publishing S.L. Obtenido de http://www.librorealidadaugmentada.com/descargas/Realidad_Aumentada_1a_Edicion.pdf.
- » Guachalla Blanco, F. (2015). Plataforma de Geolocalización de Personas Mediante Dispositivo Móvil. *Plataforma de Geolocalización de Personas Mediante Dispositivo Móvil*. La Paz, bolivia. Obtenido de https://www.academia.edu/19729128/Tesis_de_Grado.
- » Guillen Morales, R. G. (21 de Septiembre de 2015). Computación Básica. *Dispositivos Móviles*. Obtenido de <http://conceptosdecomputacionbasica.blogspot.mx/2015/09/dispositivos-moviles.html>.
- » HelloTalk Inc. . (2017). *HelloTalk* . Obtenido de HelloTalk : <https://www.hellotalk.com/#sp>.
- » IIMD Marketing Digital. (29 de Septiembre de 2016). *Que son apps, definicion y noticias*. Obtenido de <https://iimd.com/que-son-apps/>.
- » Javier Sáez, P., González, C., Soto, P., & Carnero, D. (08 de Marzo de 2017). Neosentec Augmented Reality. *Realidad virtual y realidad aumentada, ¿En qué se*

diferencian? Asturias, España. Obtenido de <http://www.neosentec.com/realidad-virtual-y-realidad-aumentada-diferencias/>.

- » Labs, S. (2009-2017). *Kanban Tool*. Obtenido de Kanban Tool: <http://kanbantool.com/es/scrumban-scrum-y-kanban>.
- » Lara Trejo, V., & Tapia Caballero, L. (29 de Marzo de 2012). Anteproyecto. San Felipe del Progreso, Estado de México, México: Tecnológico de Estudios Superiores de San Felipe del Progreso.
- » Leapp. (30 de Abril de 2012). *Historia de las Aplicaciones Móviles* .
- » Legorreta, G. (08 de Octubre de 2015). *Profesionistas.org.mx*. Obtenido de Aprendizaje Inmersivo: <http://profesionistas.org.mx/aprendizaje-inmersivo/>.
- » Lenguaje de programación JavaScript. (2006-2015). *La Revista Informática.com*. Obtenido de <http://www.larevistainformatica.com/JavaScript.htm>.
- » Luna Pérez, D. A. (2016). Métodos Aplicados Al Desarrollo De Aplicaciones Móviles. Danthop, 8. Obtenido de <http://info.danthop.com/estrategias-de-desarrollo-de-aplicaciones-moviles-0>
- » Marín, D. (6 de Abril de 2015). Sistemas Operativos Móviles. *Ciencia y Tecnología Remington*. Obtenido de <http://tecnologiaremington.esy.es/2015/04/>.
- » Martín Martín, A. O. (08 de 03 de 2012). *queaprendemoshoy.com*. Obtenido de ¿Qué es la Calidad? (VI): El Modelo ISO 9001 de Gestion de la Calidad: <http://queaprendemoshoy.com/%C2%BFque-es-la-calidad-vi-el-modelo-iso-9001-de-gestion-de-la-calidad/>.
- » Memrise. (15 de 05 de 2017). *iTunes Preview*. Obtenido de Memrise: Learn Languages: <https://itunes.apple.com/us/app/memrise-learn-languages/id635966718?mt=8>.

- » Meneses Sánchez, J., & Laveranio Meca, E. (2016). Prototipo de Aplicación Móvil utilizando la Metodología Mobile-D para la Verificación de la Formalidad en el Servicio de Taxi Metropolitano en la Ciudad de Lima. *Prototipo de Aplicación Móvil utilizando la Metodología Mobile-D para la Verificación de la Formalidad en el Servicio de Taxi Metropolitano en la Ciudad de Lima*. Lima, Perú.

- » Microsoft . (2017). *Microsoft*. Obtenido de Visual Studio Professional 2017: [https://www.microsoftstore.com/store/msmx/es_MX/pdp/productID.5095931900?gclid=CICzzOK989MCFaFKMgodwb4LJg&gclid=ds&tduid=\(8a86f7b84644572a164373402b8c7e0f\)\(230667\)\(2214473\)\(jpk_CICzzOK989MCFaFKMgodwb4LJg\)\(\)](https://www.microsoftstore.com/store/msmx/es_MX/pdp/productID.5095931900?gclid=CICzzOK989MCFaFKMgodwb4LJg&gclid=ds&tduid=(8a86f7b84644572a164373402b8c7e0f)(230667)(2214473)(jpk_CICzzOK989MCFaFKMgodwb4LJg)()).

- » Microsoft Developer. (2003). *Resumende las características de C#*. Obtenido de [https://msdn.microsoft.com/es-es/library/aa287483\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa287483(v=vs.71).aspx).

- » Moreno Gorgonio, U., & Trejo Quintero, S. (24 de Abril de 2017). Encuesta Sobre Aplicaciones Moviles (apps). *Encuesta Sobre Aplicaciones Moviles (apps)*. México, Estado de México, México. Obtenido de Encuesta Sobre Aplicaciones Moviles (apps).

- » Orjuela Duarte, A., & Rojas C., M. (2 de 06 de 2008). Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo. (UAEM REDALYC, Ed.) *Revista Avances en Sistemas e Informática*, 5(2), 15. Obtenido de <http://www.redalyc.org/html/1331/133115027022/>.

- » Patricio, B. (23 de Octubre de 2013). *IT Linux Blog Compartimos lo que aprendemos cada día*. Obtenido de ¿Que es DevOps?: <http://blog.itlinux.cl/blog/2013/10/23/que-es-devops/>.

- » Pearson Education Limited . (4 de 02 de 2014). *Google Play*. Obtenido de Phonic Tricksters: <https://play.google.com/store/apps/details?id=com.odi.phonictricksters&hl=es>.

- » Pinto, A., López de Toledo, I., & Muñoz , L. (31 de 01 de 2015). *Ciencia a la Última Blog de trabajo de los alumnos de 1° de bachillerato del Colegio Base dedicado a la asignatura de Cultura Científica*. Obtenido de Realidad Aumentada y Realidad Virtual: <http://ciencialultima.blogspot.mx/2015/01/realidad-aumentada-y-realidad-virtual.html>.

- » Posada , F. (18 de Enero de 2014). canalTIC.com *Uso educativo de las TIC. Realidad Aumentada en el aula*. Asturias, España. Obtenido de http://canaltic.com/blog/?page_id=2.

- » Pressman, R. S. (2010). *Ingeniería del Software: Un enfoque práctico*. En R. S. Pressman, *Ingeniería del Software: Un enfoque práctico*. (7ma. Edición ed., pág. 805). México: McGraw-Hill Interamericana Editores S.A. de C.V.

- » Proyectos agiles.org. (s.f.). *proyectos agiles.org*. Obtenido de Demostración de requisitos completados (Sprint Review): <https://proyectosagiles.org/demostracion-requisitos-sprint-review/>.

- » Puon, L. (31 de 03 de 2014). *Merca 2.0 MercadotecniaPublicidadMedios*. Obtenido de Los 5 tipos de usuarios móviles. : <https://www.merca20.com/los-5-tipos-de-usuarios-moviles/>.

- » Ramírez, E. (16 de 09 de 2015). *TEAMSOFT*. Obtenido de ¿Qué es Calidad del Software? Parte I: <http://www.teamsoft.com.pe/blog/que-es-calidad-del-software-parte-i/>.

- » Real Academia Española. (2014). *Diccionario de la Lengua Española*. Obtenido de Diccionario de la Lengua Española: <http://dle.rae.es/index.html>.

- » Rodlt, I. (28 de Marzo de 2012). Microsoft TechNet. *¿Qué es la ingeniería de software?* .

- » Rodríguez , T. (30 de Septiembre de 2011). *Métodos aplicables para el desarrollo de aplicaciones móviles*. GENBETA: dev, 5. Obtenido de

<https://www.genbetadev.com/desarrollo-aplicaciones-moviles/metodos-aplicables-para-el-desarrollo-de-aplicaciones-moviles>

- » Romeu, A. (9 de 10 de 2014). *El Blog del Inform@tico*. Obtenido de Planning poker- ¿Cómo planificar un sprint en Scrum?: <http://albertoromeu.com/scrum-planning-poker/>.
- » Rosillo, P. (23 de 06 de 2013). *XOMBIT Tecnología*. Obtenido de ¿Qué puede ofrecer Ubuntu para Tablets y Smartphone frente a Android?: <https://xombit.com/2013/06/ubuntu-tablets-smartphone-frente-android>.
- » Schenone Marcelo , H. (2004). Diseño de una Metodología Ágil de Desarrollo de Software. *Diseño de una Metodología Ágil de Desarrollo de Software*. Buenos Aires, Argentina. Obtenido de <http://materias.fi.uba.ar/7500/schenone-tesisdegradoingenieriainformatica.pdf>.
- » Scrum.org. (2017). *Scrum.org*. Obtenido de ¿Qué es una retrospectiva de sprint?: <https://www.scrum.org/resources/what-is-a-sprint-retrospective>.
- » Shuler, C., Winters, N., & West, M. (2013). El Futuro del Aprendizaje Móvil, Implicaciones para la Planificación y la Formulación de Políticas. *El Futuro del Aprendizaje Móvil, Implicaciones para la Planificación y la Formulación de Políticas*. UNESCO. Obtenido de www.unesco.org/cen/centro-documentacion-pdf/Aprendizaje%20movil%20UNESCO.pdf.
- » Skritter. (2017). *Google Play*. Obtenido de Skritter: <https://skritter.com/features>.
- » Sommerville, I. (2005). *Ingeniería de Software* (Septima Edición ed.). Madrid, Madrid, España: PEARSON. Obtenido de <http://zeus.inf.ucv.cl/~bcrawford/Modelado%20UML/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20lan%20Sommerville.pdf>.
- » Soria, M. (15 de 02 de 2011). *Antevenio Anticipation eMarketing*. Obtenido de Seis perfiles de usuarios de Smartphones según las aplicaciones que descargan. :

<http://www.antevenio.com/blog/2011/02/seis-perfiles-usuarios-smartphones-por-aplicaciones-descargan-movil/>.

- » Tavera Moreno, V. (25 de 01 de 2012). Calidad de Software Veronica. *Factores que determinan la calidad del software*. Obtenido de <http://calidaddesoftwareveronica.wikispaces.com/Factores+que+determinan+la+calidad+del+software>.

- » The Chairman's Bao Ltd. . (11 de 05 de 2017). *Google Play*. Obtenido de The Chairman's Bao Chinese News: <https://play.google.com/store/apps/details?id=uk.co.metapps.thechairmansbao&hl=es>.

- » UNESCO. (2017). *Organización de las Naciones Unidas para la Educación la Ciencia y la Cultura*. Obtenido de Organización de las Naciones Unidas para la Educación la Ciencia y la Cultura: <http://es.unesco.org/themes/tic-educacion>.



UAEM

Universidad Autónoma
del Estado de México

CENTRO UNIVERSITARIO UAEM TEXCOCO

“Aplicación de la Metodología Mobile-D™,
para la creación del prototipo de App
Caso de estudio: Draw-HánzÍ, sustentada en AR.”

ANEXO I

**Encuesta digital sobre el uso de las
Aplicaciones móviles (Apps) en México, al
público en general y Encuesta digital sobre el
uso de metodologías de desarrollo para
Aplicaciones Móviles (Apps), en México.**



HUMANISMO
QUE TRANSFORMA

TEXCOCO, EDO. MÉX., OCTUBRE 2017

1. ENCUESTA DIGITAL A PUBLICO EN GENERAL

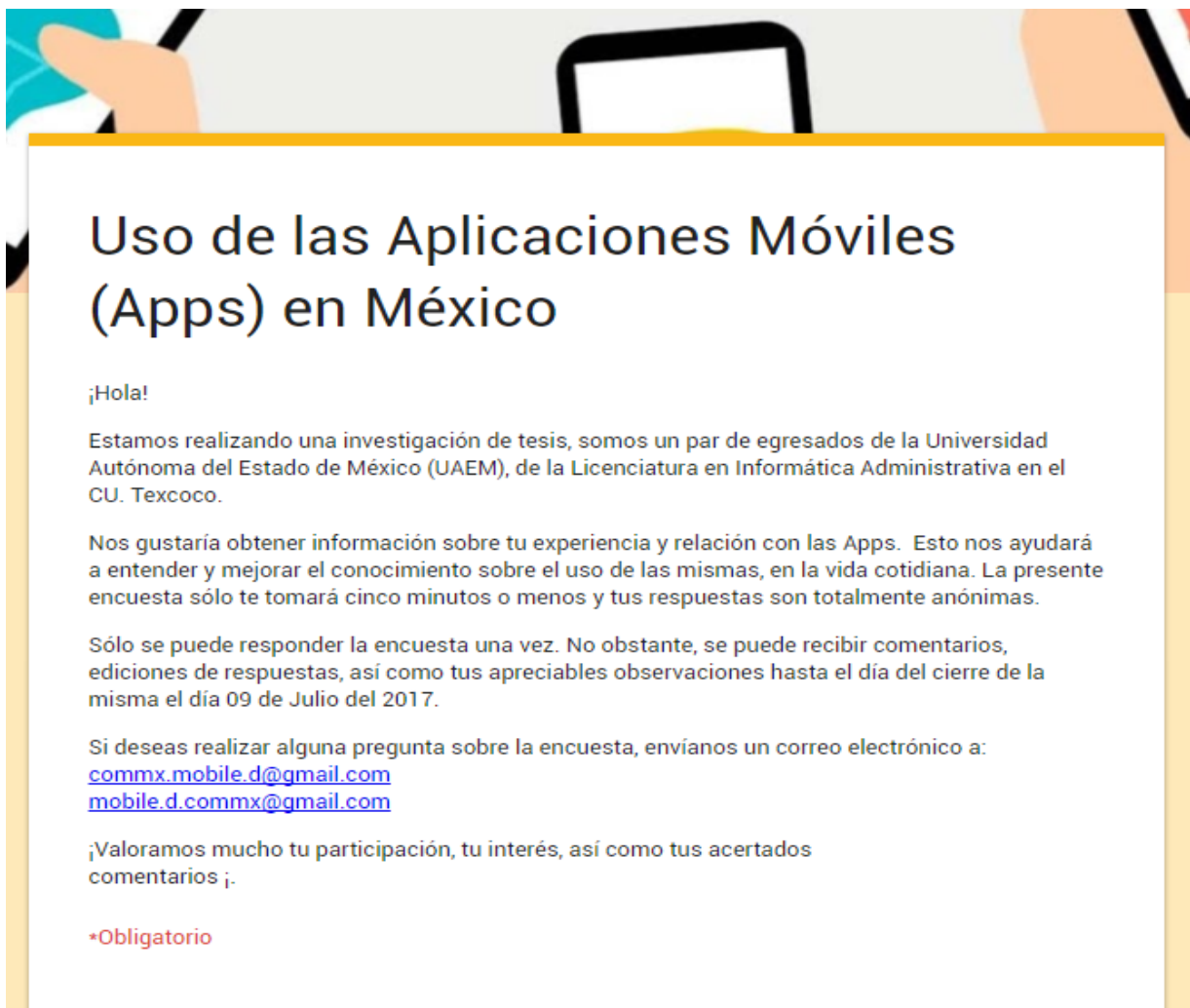
1.1. Encuesta digital sobre el uso de las aplicaciones móviles (Apps) en México.

La encuesta fue desarrollada, por medio de la plataforma Google Form y estuvo disponible para ser contestada en la siguiente dirección:

<https://goo.gl/forms/JAxVfdULaCJ8OGXi2>

Nota 1: Se presentará en formato imagen tal cual apare en la plataforma (diseño y contenido).

Figura A. Texto de presentación a la encuesta.



Fuente: Elaboración propia, 2017.

Figura B. Pregunta número uno, cantidad de dispositivos disponibles por usuario.

¿Qué dispositivos tienes en este justo momento? *
Puedes marcar más de una opción de ser necesario.

- SmartPhone (Telefono Inteligente)
- Tablet
- SmartWach
- Laptop
- Videoconsola Portátil
- Reproductor de Audio Portátil
- Asistente Digital Personal
- Otros: _____

Fuente: Elaboración propia, 2017.

Figura C. Pregunta número dos, cantidad de Apps instaladas por dispositivo disponible.

En promedio ¿Cuántas Apps tienes instaladas, en este justo momento en tu o tus dispositivos móviles? *
(p.ej. Facebook, WhatsApp, Instagram, Messenger, etc)

- De 5 a 8 Apps
- De 8 a 12 Apps
- De 12 a 15 Apps
- Más de 15 Apps

Fuente: Elaboración propia, 2017.

Figura D. Pregunta número tres, empleo diario de Apps, en los dispositivos disponibles.

¿Qué tan probable es que en un día común y corriente, emplees tu o tus dispositivos móviles, para otro fin que NO sea usar una App? *

	1	2	3	4	5	
Nada Probable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremadamente Probable

Fuente: Elaboración propia, 2017.

Figura E. Pregunta número cuatro, inversión de tiempo del usuario en Apps.

En promedio ¿Cuántas horas al día inviertes utilizando una o más Apps, en tu o tus dispositivos móviles? *

Por favor contesta sinceramente. Recuerda las respuestas son anónimas.

- Menos de 3 horas
- De 3 a 5 horas
- De 5 a 8 horas
- De 8 a 12 Horas
- De 12 a 15 Horas
- Más de 15 horas

Fuente: Elaboración propia, 2017.

Figura F. Pregunta cinco, uso promedio de las Apps por el usuario.

¿Para qué situaciones utilizas una o más Apps diariamente? *

Puedes marcar más de una opción de ser necesario.

- COMPRAS
- ENTRETENIMIENTO (p.ej. juegos, ver películas, ver T.V. etc)
- REDES SOCIALES (p. ej. Facebook, Twitter, etc)
- COMUNICACIÓN (p.ej. WhatsApp, Messenger, Telgram, etc)
- LECTURA (p.ej. Periódicos, Libros, Revistas, etc)
- TRANSPORTE (p.ej. Uber, Cabify, TaxiAmigo, etc)
- RESERVACIONES (p.ej. Restaurantes, Conciertos, Obras de Teatro, etc)
- APRENDIZAJE (p.ej. Udemy, Dualingo, Cousera, etc)
- BÚSQUEDA DE OFERTAS LABORALES (p.ej. Linkedin, OCC, etc)
- EDICIÓN (p.ej. PicsArt, Pixlr, etc }
- FOTOGRAFÍA (p.ej. Instagram, Snapchat, etc)
- MANUALIDADES (p.ej. Pinterest, Tumblr, etc)
- REALIZAR UNA TAREA ESPECÍFICA (p.ej. Decoración de Interiores, Gestión de Tareas, Aumentar productividad, etc)
- MI TRABAJO (son Apps, diseñadas por la empresa donde trabajo y con ellas desempeño mis tareas)
- AUDIO (p. ej, Spotify, Radio, etc)
- Ninguna de las Ateriores
- Otros: _____

Fuente: Elaboración propia, 2017.

Figura G. Pregunta número seis, características positivas en una App.

Con base a tu experiencia en el uso de Apps. ¿Cómo definirías una App de "Alta o Excelente Calidad" ? *

Texto de respuesta largo

Fuente: Elaboración propia, 2017.

Figura H. Pregunta número siete, características negativas en una App.

Para ti ¿Qué características tiene una Mala o Pésima App? *

Incluye todas las que consideres necesarias.

Tu respuesta

Fuente: Elaboración propia, 2017.

Figura I. Pregunta número ocho, factor crucial para descargar una App.

Para ti ¿Cuál es el factor de mayor peso, al decidir positivamente descargar y/o instalar una App, en tu o tus dispositivos móviles? *

Tu respuesta

Fuente: Elaboración propia, 2017.

Figura J. Pregunta número nueve, medio de descarga de Apps por parte del usuario.

Generalmente ¿Por qué medio consigues tus Apps? *

- AppStores (p.ej. Google Play, iTunes, etc)
- .APK (p. ej. el desarrollador me la envía personalmente)
- .IPA (p. ej. el desarrollador me la envía personalmente)
- Otros: _____

Fuente: Elaboración propia, 2017.

Figura K. Pregunta número diez, publicidad efectiva para una App.

Con base a tu experiencia ¿Cuál es el mejor medio para realizar una publicidad eficiente a una App nueva? *

Tu respuesta _____

SIGUIENTE

Nunca envíes contraseñas a través de Formularios de Google.

Google no creó ni aprobó este contenido. Denunciar abuso - Condiciones del servicio - Condiciones adicionales

Google Formularios

Fuente: Elaboración propia, 2017.

Figura I. Mensaje de despedida y envío de los resultados de la misma a la base de datos.

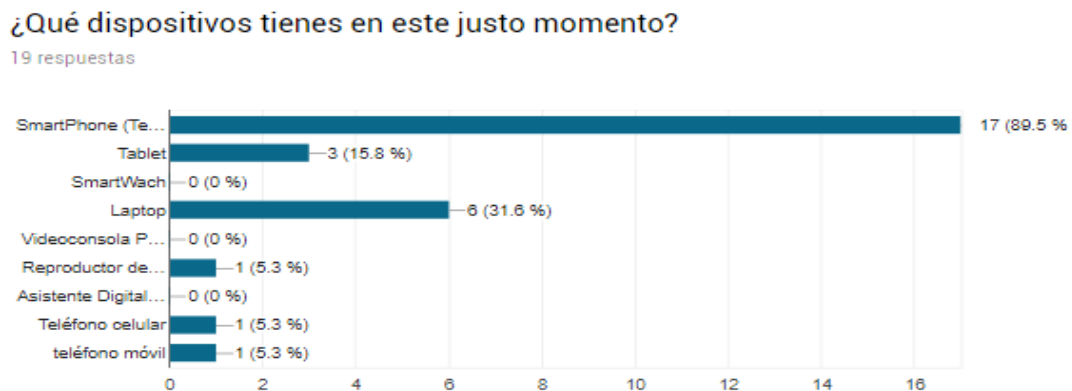


Fuente: Elaboración propia, 2017.

1.1.2. Respuesta de la encuesta sobre el uso de las aplicaciones móviles (Apps) en México.

El mundo muestra es de 19 personas, mismas que contestaron la encuesta al día 09 de Julio del 2017 a las 24:00 Hrs (hora del cierre de la misma), arrojando los siguientes resultados:

Figura a. Gráfica de respuestas para la pregunta número uno.

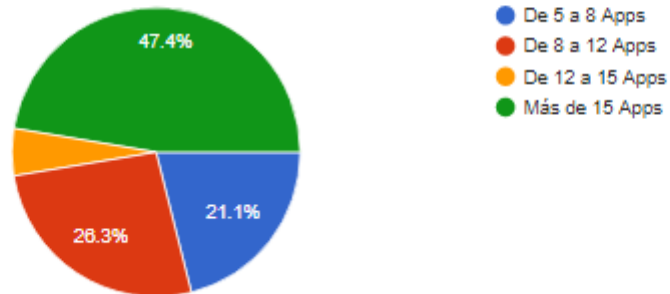


Fuente: Elaboración propia, 2017.

Figura b. Gráfica de respuestas para la pregunta número dos.

En promedio ¿Cuántas Apps tienes instaladas, en este justo momento en tu o tus dispositivos móviles?

19 respuestas

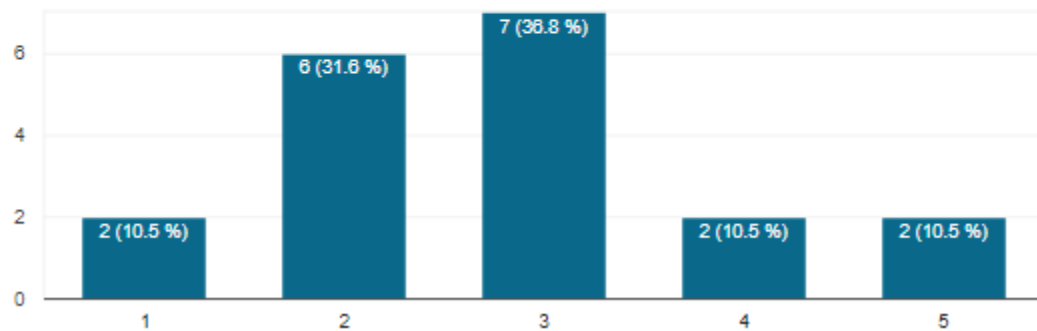


Fuente: Elaboración propia, 2017.

Figura c. Gráfica de respuestas para la pregunta número tres.

¿Qué tan probable es que en un día común y corriente, emplees tu o tus dispositivos móviles, para otro fin que NO sea usar una App?

19 respuestas

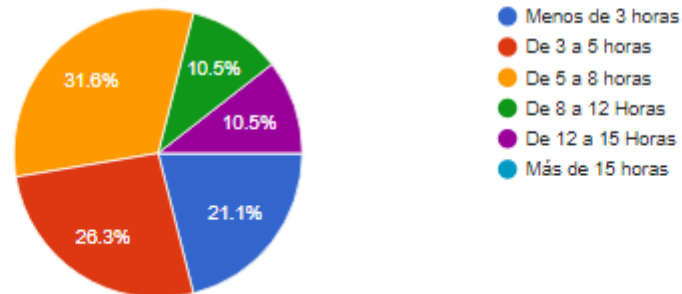


Fuente: Elaboración propia, 2017.

Figura d. Gráfica de respuestas para la pregunta número cuatro.

En promedio ¿Cuántas horas al día inviertes utilizando una o más Apps, en tu o tus dispositivos móviles?

19 respuestas

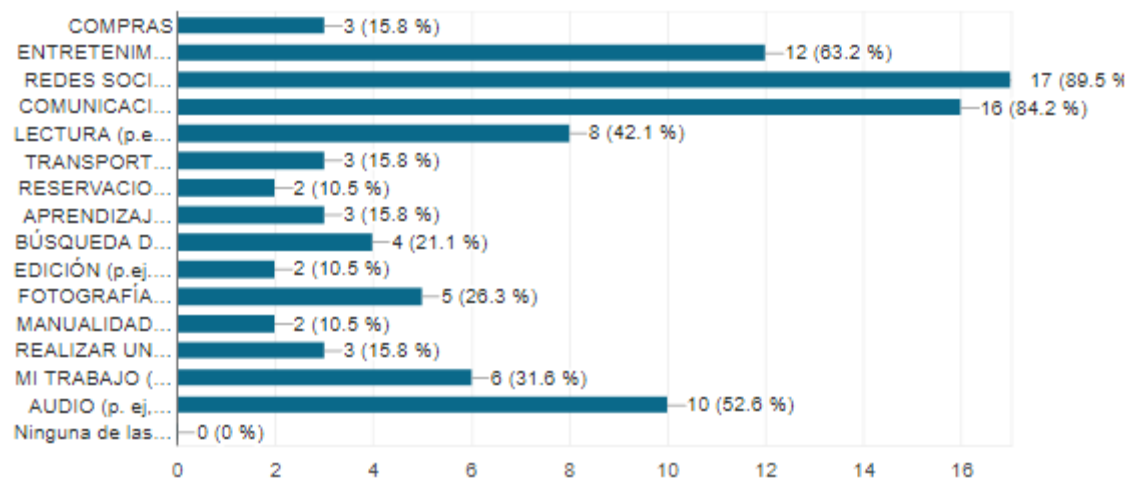


Fuente: Elaboración propia, 2017.

Figura e. Gráfica de respuestas para la pregunta número cinco.

¿Para qué situaciones utilizas una o más Apps diariamente?

19 respuestas

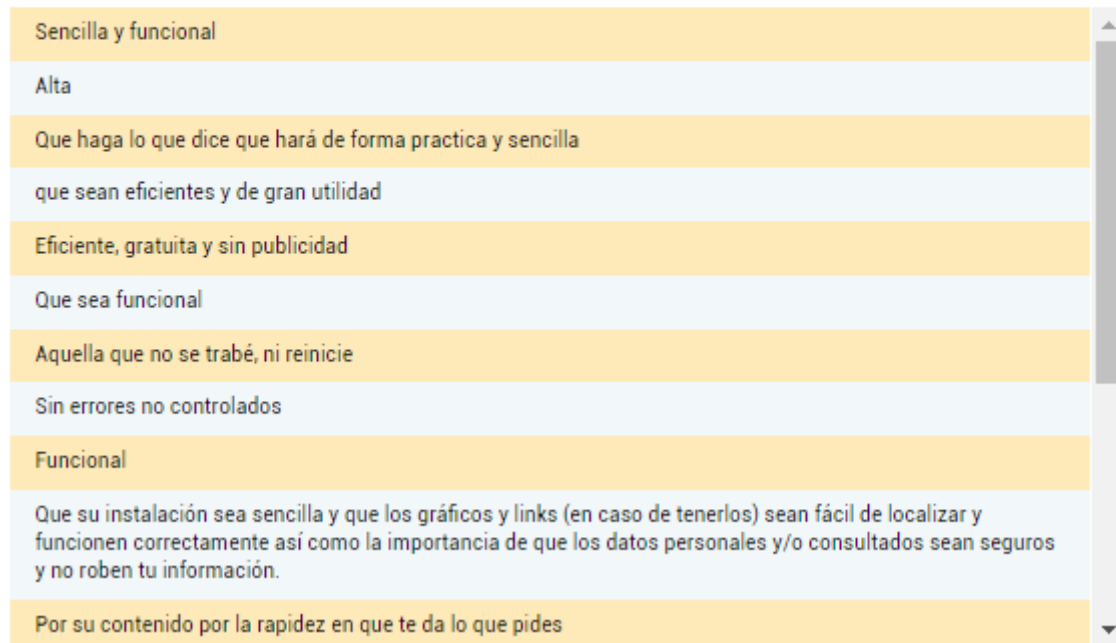


Fuente: Elaboración propia, 2017.

Figura f. Parte I de la lista de respuestas para la pregunta número seis.

Con base a tu experiencia en el uso de Apps. ¿Cómo definirías una App de "Alta o Excelente Calidad" ?

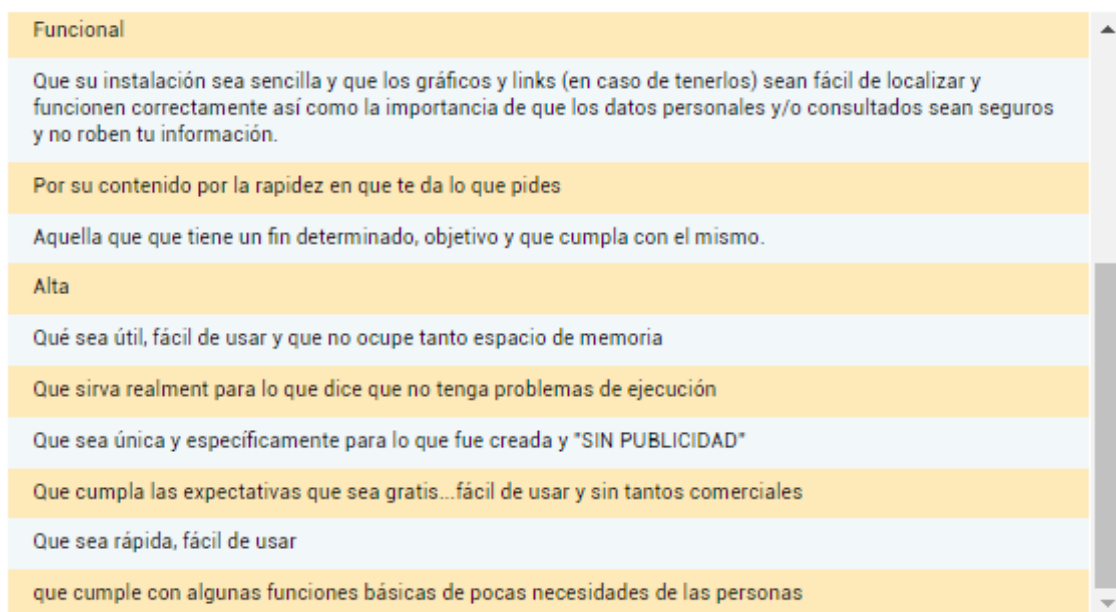
19 respuestas



Sencilla y funcional
Alta
Que haga lo que dice que hará de forma practica y sencilla
que sean eficientes y de gran utilidad
Eficiente, gratuita y sin publicidad
Que sea funcional
Aquella que no se trabé, ni reinicie
Sin errores no controlados
Funcional
Que su instalación sea sencilla y que los gráficos y links (en caso de tenerlos) sean fácil de localizar y funcionen correctamente así como la importancia de que los datos personales y/o consultados sean seguros y no roben tu información.
Por su contenido por la rapidez en que te da lo que pides

Fuente: Elaboración propia, 2017.

Figura g. Parte II de la lista de respuestas para la pregunta número seis.



Funcional
Que su instalación sea sencilla y que los gráficos y links (en caso de tenerlos) sean fácil de localizar y funcionen correctamente así como la importancia de que los datos personales y/o consultados sean seguros y no roben tu información.
Por su contenido por la rapidez en que te da lo que pides
Aquella que que tiene un fin determinado, objetivo y que cumpla con el mismo.
Alta
Qué sea útil, fácil de usar y que no ocupe tanto espacio de memoria
Que sirva realment para lo que dice que no tenga problemas de ejecución
Que sea única y específicamente para lo que fue creada y "SIN PUBLICIDAD"
Que cumpla las expectativas que sea gratis...fácil de usar y sin tantos comerciales
Que sea rápida, fácil de usar
que cumple con algunas funciones básicas de pocas necesidades de las personas

Fuente: Elaboración propia, 2017.

Figura h. Parte I de la lista de respuestas para la pregunta número siete.

Para ti ¿Qué características tiene una Mala o Pésima App?

19 respuestas

Con exceso de publicidad
Qué no cumpla con su función
Mucha Publicidad y que este muy complicada de utilizar
que pida muchas cosas o espacio para instalar , que alenten las demás aplicaciones.
Anuncios, uso excesivo de memoria ram
Que no funcione correctamente
Aquella que no se trabé, ni reinicie
Que no cumpla con su descripción
No sirve, te hecha a perder el entusiasmo
Que no se pueda instalar en todas las versiones de sistema operativo, que dependiendo de las versiones los gráficos sean malos, que utilice tus datos personales para mal uso, que consuma mucha batería.
Sus gráficos
tener muchas opciones y no funcionar correctamente alguna o todas, ser lenta y cierres inesperados.

Fuente: Elaboración propia, 2017.

Figura i. Parte II de la lista de respuestas para la pregunta número siete.

Sus gráficos
tener muchas opciones y no funcionar correctamente alguna o todas, ser lenta y cierres inesperados.
Mala
Que no tenga utilidad
Problemas al ejecutar, no hace correctamente para lo que es. Muy ambigua, errores al utilizarla.
1.- Que sea lenta o se trabe 2.-Que no cumpla con las especificaciones que fue diseñada 3.-Que redireccione a otros sitios que no existen 4.-Que necesite de otra app para que se pueda utilizar 5.-Que no sea entendible o fácil de manejar
Que consume mucha memoria.. No es intuitiva y no cumple con su función
Cuando se trava frecuentemente, cuando carga lentamente, cuando solo se debe instalar en la memoria del teléfono y no se puede en la memoria externa
que contiene mucho anuncios y muy malas funciones

Fuente: Elaboración Propia, 2017.

Figura j. Parte I de la lista de respuestas para la pregunta número ocho.

Para ti ¿Cuál es el factor de mayor peso, al decidir positivamente descargar y/o instalar una App, en tu o tus dispositivos móviles?

19 respuestas

Si lo necesito verdaderamente
Que sea lo necesito
Las buenas referencias
ambas
El costo y el tamaño
popularidad
Los virus
La calificacion de otros usuarios
Que sea efectivo en razón de sus funciones
Que sean seguras, que no tengan virus
las redes sociales
La necesidad que tenga en el momento y si la app cumple con lo que busco. (ademas de leer las criticas)

Fuente: elaboración propia, 2017.

Figura k. Parte II de la lista de respuestas para la pregunta número ocho.

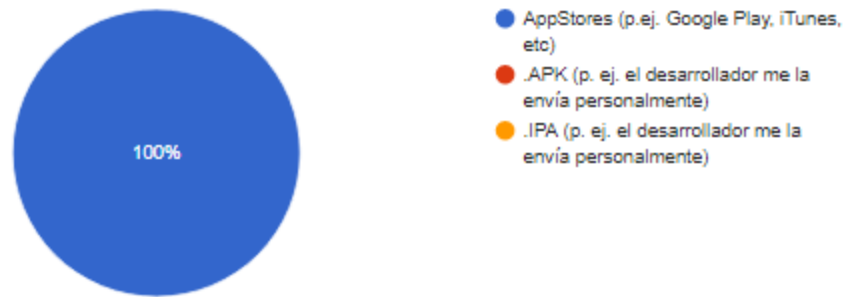
La calificacion de otros usuarios
Que sea efectivo en razón de sus funciones
Que sean seguras, que no tengan virus
las redes sociales
La necesidad que tenga en el momento y si la app cumple con lo que busco. (ademas de leer las criticas)
comentario que ya han usado el app
Que cubra alguna necesidad
Que me sirva para lo que necesito y la calificación
La calidad de la app, si la descargo y no me sirve la desecho
Que me aporte valor al disminuir mi tiempo de trabajo
Que no utilice mucha memoria del teléfono
que se instaló correctamente al dispositivo movil

Fuente: Elaboración propia, 2017.

Figura I. Gráfica de respuestas para la pregunta número nueve.

Generalmente ¿Por qué medio consigues tus Apps?

19 respuestas



Fuente: Elaboración propia, 2017.

Figura m. Parte I de la lista de respuestas para la pregunta número diez.

Con base a tu experiencia ¿Cuál es el mejor medio para realizar una publicidad eficiente a una App nueva?

19 respuestas



Fuente: Elaboración propia, 2017.

Figura n. Parte II de la lista de respuestas para la pregunta número diez.

Redes sociales
Online
Que sea anunciada en las redes sociales más comunes (Twitter, Snapchat, Facebook, Instagram, etc)
un video mostrando su contenido
Redes sociales y paginas de tecnología.
Red social
Yo me he enterado de ellas en programas especializados que transmiten en la radio
Otras apps
Dando demostraciones interactivas
Mediante redes sociales, porque hay difusión.
Por redes sociales
los medios de comunicación (radio, televisión, etc)

Fuente: Elaboración propia, 2017.

1.1.3. Interpretación de las respuestas para la encuesta sobre el uso de las aplicaciones móviles (Apps) en México.

Los resultados de la encuesta, para conocer los usos y costumbres del usuario de Apps en México y que con base en los datos anteriores se interpreta de la siguiente manera:

El 57.89% de la población muestra tiene en promedio tres dispositivos móviles, a su disposición, siendo estos: un Smartphone, una Tablet y una laptop, en los que tienen instaladas más de 15 Apps, ya sea en lo particular y/o universal, donde es medianamente probable ser utilizados para no usar una App. Sin embargo, el 31.6% de la población muestra, invierte un promedio de 5 a 8 horas en el uso de Apps al día para las siguientes actividades: donde destaca el monitoreo de sus redes sociales con un 89.5%, seguido de la comunicación con un 84.2%, finalizando con el 63.2% de la población muestra utilizándolas para entretenimiento; *lo que dejó muy por debajo de la media con un 15.8% de la población muestra interesada en el aprendizaje y/o educación, por medio de una App.*

No obstante, la población muestra señala que para que una App sea considerada de “Alta y/o Excelente calidad”, esta deberá contener las siguientes características: “deberá ser funcional debido a que hará lo que promete hacer, eficiente debido a que lo hará con la rapidez justa para su correcto funcionamiento, sencilla y practica debido a que los gráficos serán fáciles de localizar, además de garantizar la seguridad e integridad de los datos. No obstante, se hace un énfasis a nuestro punto de vista irracional, debido a que se exige sea *gratuita, sin publicidad y con un bajo consumo de memoria (determinadas como situaciones de riesgo para depreciar la calidad de la misma App)*”, lo que imposibilita el desarrollo de las Apps con el fin de generar un lucrativo negocio. No obstante, para el desarrollo sin ánimo de lucro, siendo más bien como complemento de un negocio y/o institución, si tendría gran demanda y aceptación. A causa de lo anterior es como se determinan que la necesidad y las buenas calificaciones que tenga la App, son de vital importancia para generar la motivación necesaria para su descarga e instalación. Determinando sin lugar a dudas que el medio de alcance son las AppStore’s propias de los OS, sugiriendo que el mejor medio de publicidad es por medio de las redes sociales.

1.2. Encuesta digital sobre el uso de metodologías de desarrollo para Apps, en México.

La encuesta fue desarrollada, por medio de la plataforma Google Form y estuvo disponible para ser contestada en la siguiente dirección:

<https://goo.gl/forms/Nly0KvQNrViN6sDR2>

Nota 2: Se presentará en formato imagen tal cual apare en la plataforma (diseño y contenido).

Figura J. Texto de presentación de la encuesta.



Fuente: Elaboración propia, 2017.

Figura K. Pregunta número uno, para determinar el tipo de desarrollador.

The image shows a screenshot of a survey question. The question is "¿Qué tipo de desarrollador eres? *". Below the question, there are four radio button options: "Freelance", "Empresa Desarrolladora de Software (p.ej. dueño, empleado...)", "Ambos", and "Otros: _____".

Fuente: Elaboración propia, 2017.

Figura L. Pregunta número dos, para determinar los años de experiencia.

¿Cuántos años de experiencia tienes en el desarrollo de software ? *

- Menos de 3 años
- De 3 a 6 años.
- De 6 a 9 años.
- De 9 a 12 años.
- Más de 12 años.

Fuente: Elaboración propia, 2017.

Figura M. Pregunta número tres, definir los roles que ha desempeñado.

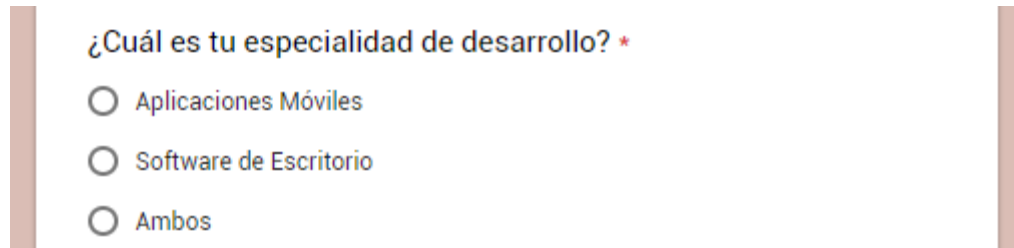
¿Qué tipo de puesto has desempeñado?, en los proyectos de desarrollo en los que has participado. *

Se puede marcar más de una casilla en caso de ser necesario.

- Consultor.
- Gerente de Proyecto.
- Visionario.
- Analista.
- Diseñador.
- Arquitecto de software.
- Arquitecto de Sistema.
- Jefe de Desarrolladores.
- Desarrollador de software.
- Programador de software.
- Administrador de código.
- Jefe de Producto.
- Responsable de SQA.
- Tester.
- Administrador de cuentas.
- Administrador de Sistemas.
- Capacitador.

Fuente: Elaboración propia, 2017.

Figura N. Pregunta número cuatro, definir experiencia de desarrollo.



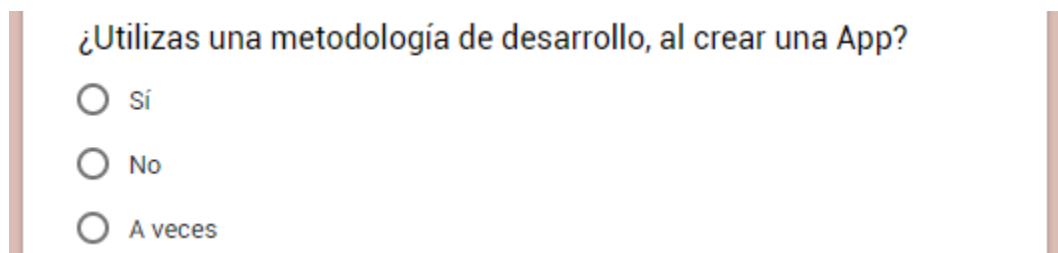
¿Cuál es tu especialidad de desarrollo? *

- Aplicaciones Móviles
- Software de Escritorio
- Ambos

Fuente: Elaboración propia, 2017.

Nota 3: Dependiendo de la respuesta de la pregunta número cuatro direccionara a una sección diferente. Mismas que se mostraran por orden de consecución, como se muestran en las respuestas.

Figura Ñ. Pregunta número 4.1, determinar el uso de las metodologías para Apps.

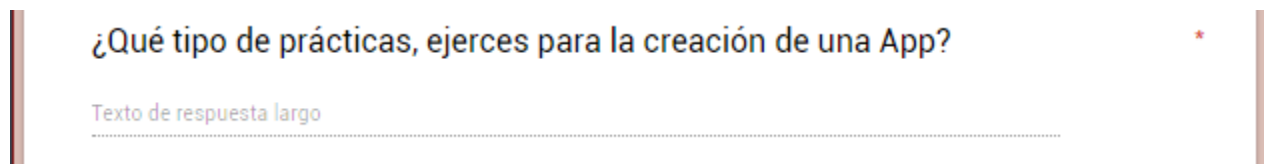


¿Utilizas una metodología de desarrollo, al crear una App?

- Sí
- No
- A veces

Fuente: Elaboración propia, 2017.

Figuras O. Pregunta número 4.2, determinar las ventajas no utilizar una metodología.



¿Qué tipo de prácticas, ejerces para la creación de una App? *

Texto de respuesta largo

Fuente: Elaboración propia, 2017.

Figuras P. Pregunta número 4.3, determinar las ventajas de utilizar una metodología.

¿Por qué tipo de situación, consideras que un desarrollador (independientemente de su tipo, p.ej. Freelance y/o empresa), NO sigue una metodología de desarrollo? *

Texto de respuesta largo

Fuente: Elaboración propia, 2017.

Figura Q. Pregunta número 4.4, determinar las ventajas de no utilizar una metodología.

Desde tu punto de vista, ¿Cómo definirías la "Alta Calidad" de una App *

Texto de respuesta largo

Fuente: Elaboración propia, 2017.

Figura R. Pregunta número 4.5, determinar tipo de metodologías que se utilizan.

¿Utilizas metodologías de desarrollo para software?

- Tradicionales
- Ágiles
- Híbridos
- Mixtas
- Ambas, pero por separado. Depende del proyecto
- Otros: _____

Fuente: Elaboración propia, 2017.

Figura S. Pregunta número 4.6, ventajas del uso de las metodologías preferidas. (En caso de contestar cualquier otra opción diferente a “ágiles”).

¿Qué ventaja(s) le atribuyes a este tipo de metodologías?

Tu respuesta

Fuente: Elaboración propia, 2017.

Figura T. Pregunta número 4.7, situación para no usar una metodología.

¿Por qué tipo de situación, consideras que un desarrollador (independientemente de su tipo, p.ej. Freelance y/o empresa), NO sigue una metodología de desarrollo? *

Tu respuesta

Fuente: Elaboración propia, 2017.

Figura U. Pregunta número 4.8, definición de alta calidad en el Software.

Desde tu punto de vista, ¿Cómo definirías la "Alta Calidad" en un software? *

Tu respuesta

Fuente: Elaboración propia, 2017.

Figura V. Pregunta número 4.9, si la respuesta a la pregunta 4.5 en la opción “Agiles”.

¿Te consideras experto o conocedor en la implementación, de las metodologías ágiles de desarrollo para Apps? *

Sí

No

Tal vez

Fuente: Elaboración propia, 2017.

Figura W. Pregunta número 4.10, si la respuesta a la pregunta 4.5 en la opción “Agiles”.

Con base, a tu experiencia de la siguiente lista ¿Cuáles consideras que son los beneficios aportados por el uso de una metodología ágil en el desarrollo de una App? *

- Calidad mejorada
- Retroalimentación continua con el cliente
- Flexibilidad en el proceso y las definiciones de los productos de software
- Entornos orientados al trabajo en equipo
- Entregas oportunas del producto
- Interacción constante
- Rápida respuesta a los cambios
- Eliminación de tareas innecesarias
- Clarificación de los requisitos del cliente, conforme avanza el proyecto
- Intervención efectiva del cliente, en el proceso de desarrollo
- Ninguna de las anteriores
- Otros: _____

Fuente: Elaboración propia, 2017.

Figura X. Pregunta número 4.11, si la respuesta a la pregunta 4.5 en la opción “Agiles”.

¿Cuáles de las siguientes metodologías ágiles para Apps, haz escuchado últimamente?

- SXP-J2ME
- DELFROID
- DMAMC
- MADAMDM
- MOBILE-D
- MOBILE DEVELOPMENT PROCESS SPIRAL
- METODOLOGÍA DE DISEÑO HÍBRIDO
- KANBAN
- SCRUM
- DevOps
- SCRUMBAN
- NINGUNA DE LAS ANTERIORES
- TODAS LAS ANTERIORES
- Otros: _____

Fuente: Elaboración propia, 2017.

Figura Y. Pregunta número 4.12, si la respuesta a la pregunta 4.5 en la opción “Agiles”.

A tu parecer ¿Cuál es la mejor metodología para desarrollar Apps? y ¿Por qué? *

Tu respuesta

Fuente: Elaboración propia, 2017.

Figura Z. Pregunta número 4.13, si la respuesta a la pregunta 4.5 en la opción “Agiles”.

¿Consideras que la implementacion de una metodología ágil de desarrollo para Apps, aporta competitividad a tu profesión y/o empresa? y ¿Por qué? *

Tu respuesta

Fuente: Elaboración propia, 2017.

Figura AA. Pregunta número 4.14, si la respuesta a la pregunta 4.5 en la opción “Agiles”.

¿Por qué tipo de situación, consideras que un desarrollador (independientemente de su tipo, p.ej. Freelance y/o empresa), NO sigue una metodología de desarrollo en la creación de una App? *

Tu respuesta

Fuente: Elaboración propia, 2017.

Figura BB. Pregunta número 4.14, si la respuesta a la pregunta 4.5 en la opción “Agiles”.

Desde tu punto de vista, ¿Cómo definirías la "Alta Calidad" de una App *

Tu respuesta

Fuentes: Elaboración propia, 2017.

Figura CC. Pregunta número 4.15, si la respuesta a la pregunta cuatro en la opción “Software de Escritorio”.

¿Utilizas una metodología de desarrollo, al crear un un producto de software?

- Sí
- No
- Talvez

Fuente: Elaboración propia, 2017.

Figura DD. Pregunta número 4.16, si la respuesta a la pregunta 4.15 en la opción afirmativamente.

¿Utilizas metodologías de desarrollo para software?

- Tradicionales
- Ágiles
- Híbridos
- Mixtas
- Ambas, pero por separado. Depende del proyecto
- Otros: _____

Fuente: Elaboración propia, 2017.

Figura EE. Pregunta número 4.17, si la respuesta a la pregunta 4.16 es cualquier opción.

¿Qué ventaja(s) le atribuyes a este tipo de metodologías?

Tu respuesta

Fuente: Elaboración propia, 2017.

Figura FF. Pregunta número 4.18, si la respuesta a la pregunta 4.16 es cualquier opción.

¿Por qué tipo de situación, consideras que un desarrollador (independientemente de su tipo, p.ej. Freelance y/o empresa), NO sigue una metodología de desarrollo? *

Tu respuesta

Fuente: Elaboración propia, 2017.

Figura GG. Pregunta número 4.19, si la respuesta a la pregunta 4.16 es cualquier opción.

Desde tu punto de vista, ¿Cómo definirías la "Alta Calidad" en un software? *

Tu respuesta

Fuente: Elaboración propia, 2017.

Figura HH. Texto de despedida y envío de respuestas a la base de datos.



Fuente: Elaboración propia, 2017.

1.2.1. Respuesta de la encuesta sobre el uso de metodologías de desarrollo para Apps, en México.

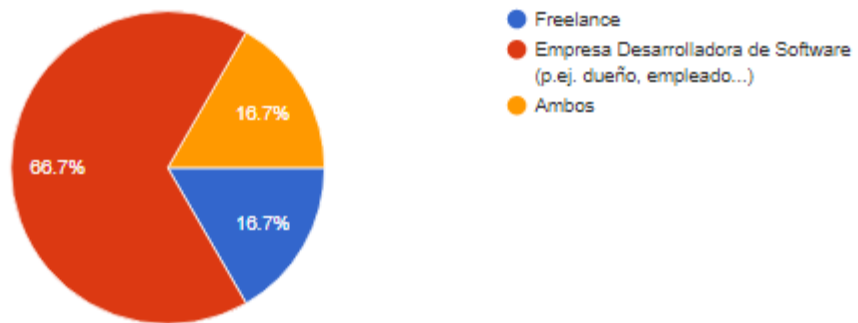
El mundo muestra es de 6 desarrolladores y/o expertos, mismos que contestaron la encuesta al día 09 de Julio del 2017 a las 24:00 Hrs (hora del cierre de la misma), arrojando los siguientes resultados:

Nota 4: Para fines prácticos solo se presentarán, las preguntas que obtuvieron una respuesta, (las que no obtuvieron ninguna respuesta se omitirán de la presentación).

Figura ñ. Gráfica de respuestas para la pregunta número uno.

¿Qué tipo de desarrollador eres?

6 respuestas

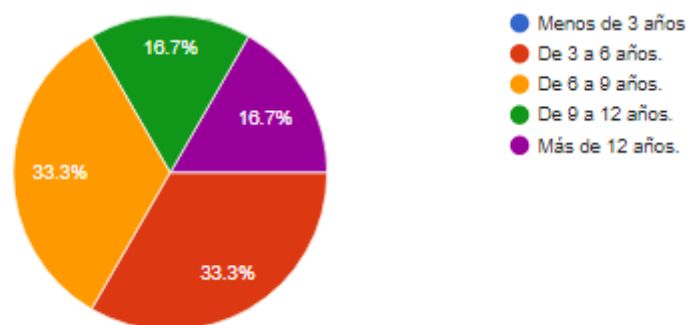


Fuente: Elaboración propia, 2017.

Figura o. Gráfica de respuestas para la pregunta número dos.

¿Cuántos años de experiencia tienes en el desarrollo de software ?

6 respuestas

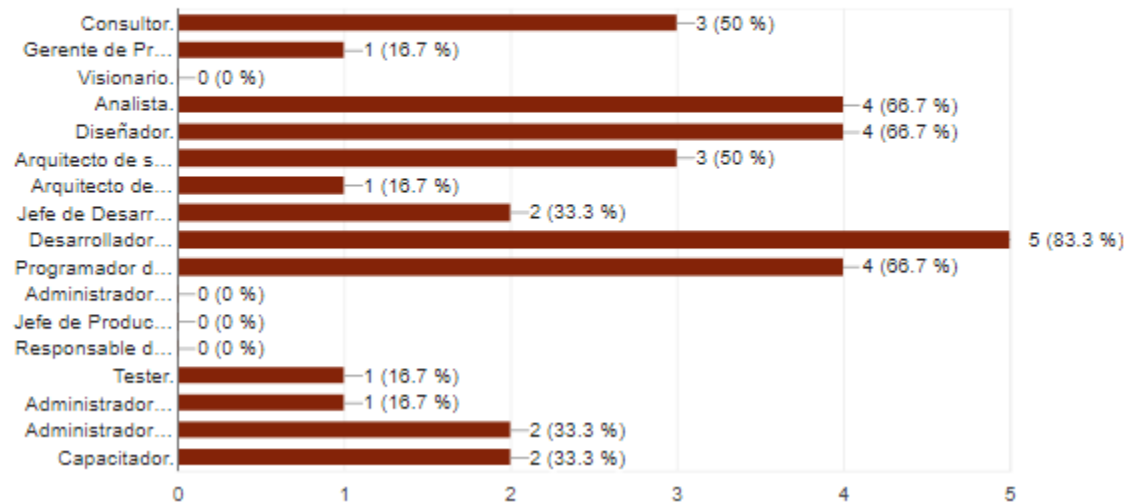


Fuente: Elaboración propia, 2017.

Figura p. Gráfica de respuestas para la pregunta número tres.

¿Qué tipo de puesto has desempeñado?, en los proyectos de desarrollo en los que has participado.

6 respuestas

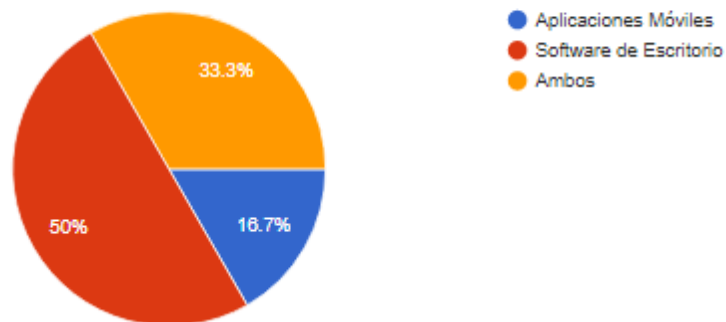


Fuente: Elaboración propia, 2017.

Figura q. Gráfica de respuestas para la pregunta número cuatro.

¿Cuál es tu especialidad de desarrollo?

6 respuestas



Fuente: Elaboración propia, 2017.

Figura r. Gráfica de respuestas para la pregunta número 4.1.

¿Utilizas una metodología de desarrollo, al crear una App?

3 respuestas

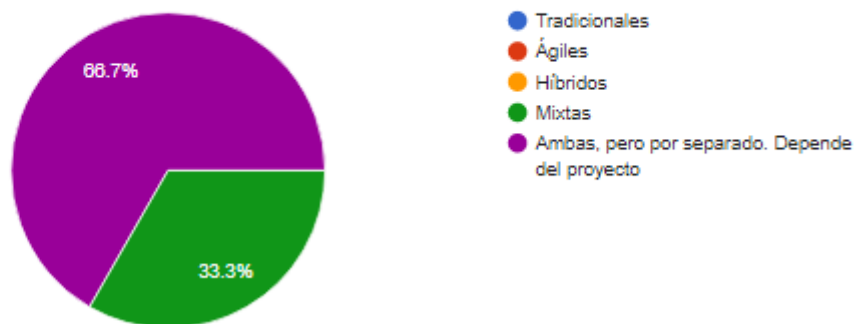


Fuente: Elaboración propia, 2017.

Figura s. Gráfica de respuestas para la pregunta número 4.5.

¿Utilizas metodologías de desarrollo para software?

3 respuestas



Fuente: Elaboración propia, 2017.

Figura t. Lista de respuestas para la pregunta 4.6.

¿Qué ventaja(s) le atribuyes a este tipo de metodologías?

1 respuesta

Las metodologías ágiles permiten la participación y colaboración del equipo de acuerdo a sus capacidades. Las metodologías tradicionales aunque son lentas permiten que el desarrollo del producto se pueda realizar por fases y presentar solo una parte de éste totalmente funcional.

Fuente: Elaboración propia, 2017.

Figura u. Lista de respuestas para la pregunta 4.7.

¿Por qué tipo de situación, consideras que un desarrollador (independientemente de su tipo, p.ej. Freelance y/o empresa), NO sigue una metodología de desarrollo?

3 respuestas

Falta de tiempo

Probablemente por desconocimiento o por falta de tiempo al tener varios proyectos a la vez.

En ocasiones el usuario requiere una solución rápida y eso implica que no haya tiempo de seguir una metodología

Fuente: Elaboración propia, 2017.

Figura v. Lista de respuestas para la pregunta 4.8.

Desde tu punto de vista, ¿Cómo definirías la "Alta Calidad" en un software

3 respuestas

0 errores no controlados y facilidad de uso por el usuario

Aquel que satisface completamente las necesidades del usuario utilizando recursos mínimos, que es eficiente y eficaz

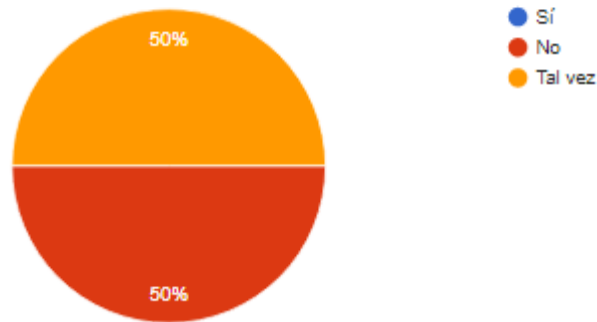
Que la aplicación realice eficientemente las tareas para las que fue diseñado, con la menor cantidad de incidencias

Fuente: Elaboración propia, 2017.

Figura w. Gráfica de las respuestas para la pregunta 4.9.

¿Te consideras experto o conocedor en la implementación, de las metodologías ágiles de desarrollo para Apps?

2 respuestas

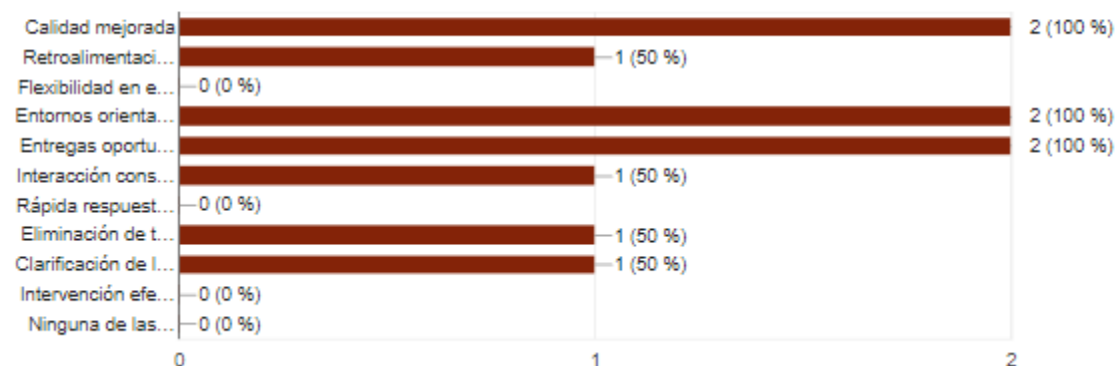


Fuente: Elaboración propia, 2017.

Figura x. Gráfica de las respuestas para la pregunta 4.10.

Con base, a tu experiencia de la siguiente lista ¿Cuáles consideras que son los beneficios aportados por el uso de una metodología ágil en el desarrollo de una App?

2 respuestas

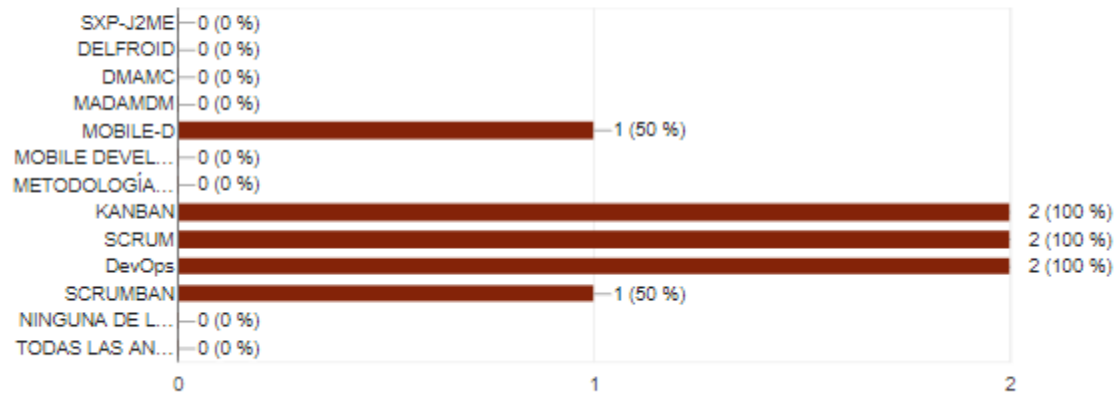


Fuente: Elaboración propia 2017.

Figura y. Gráfica de las respuestas para la pregunta 4.11.

¿Cuáles de las siguientes metodologías ágiles para Apps, haz escuchado últimamente?

2 respuestas

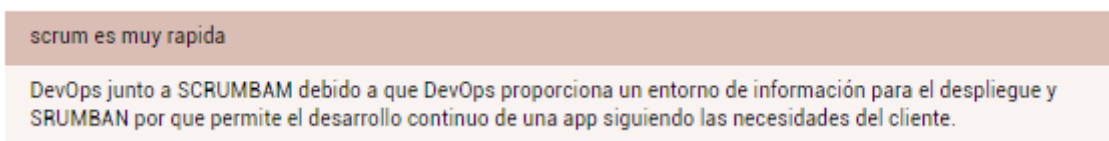


Fuente: Elaboración propia, 2017.

Figura z. Lista de respuestas para la pregunta 4.12.

A tu parecer ¿Cuál es la mejor metodología para desarrollar Apps? y ¿Por qué?

2 respuestas



Fuente: Elaboración propia, 2017.

Figura aa. Lista de respuestas para la pregunta 4.13.

¿Consideras que la implementación de una metodología ágil de desarrollo para Apps, aporta competitividad a tu profesión y/o empresa? y ¿Por qué?

2 respuestas

si por ayuda a realizar las cosas con estándares de calidad

Si, por que el uso de una metodología quiere decir que el desarrollo lleva un proceso administrativo.

Fuente: Elaboración propia, 2017.

Figura bb. Lista de respuestas para la pregunta 4.14.

¿Por qué tipo de situación, consideras que un desarrollador (independientemente de su tipo, p.ej. Freelance y/o empresa), NO sigue una metodología de desarrollo en la creación de una App?

2 respuestas

desconocimiento

Debido a la falta de conocimiento en estas o de experiencia al usarlas...

Fuente: Elaboración propia, 2017.

Figura cc. Lista de repuestas para la pregunta 4.15.

Desde tu punto de vista, ¿Cómo definirías la "Alta Calidad" de una App

2 respuestas

que hace lo que dice que hará

Métricas de uso, errores, buena documentación del código y de los procesos para su uso.

Fuente: Elaboración propia, 2017.

1.2.3. Interpretación de las respuestas a la encuesta sobre el uso de metodologías de desarrollo para Apps, en México.

Se determina que el 66.7% de nuestra población muestra pertenece al sector Startup y/o PyME IT. Con una media de experiencia labora de entre 3 a 9 años, mismos en los que se han desempeñado en los siguientes roles relevantes: consultor con un 50%, Analista con un 66.7%, diseñador con un 66.7%, Desarrollador con un 83.3%, finalmente como programador 66.7%, por mencionar algunos. No obstante, el campo de desarrollo en el sector de las Apps es solo de un 16.7%, sobrepasado por el desarrollo de software de escritorio con un 50%. A un cuando el 100% maneja una metodología de desarrollo de software y/o Apps (tradicionales, ágiles y/o mixtas), se considera No expertos, ni conocedores en dichas metodologías ágiles, se reconocen como las más utilizadas las siguientes: Kanban, Scrum, DevOps y muy por debajo en uso están Srumban y Mobile-D™.

Sin embargo, se reconocen como beneficios de implementar una metodología ágil las siguientes características: calidad mejorada, retroalimentación continua con el cliente, entornos orientados al trabajo en equipo, entregas oportunas del producto, interacción constante, eliminación de tareas innecesarias y por último clarificación de los requisitos del cliente, conforme avanza el proyecto.

La mayoría de la población muestra reconoce que el desuso de una metodología ágil de desarrollo de software para Apps, se atribuye al desconocimiento u la falta de experiencia de uso, además del desconocimiento sobre los beneficios sobre el proceso administrativo del desarrollo y de los múltiples estándares de calidad que ofrece como son: métricas de uso. Justa y eficaz documentación del código y de los procesos de desarrollo del mismo proyecto.



UAEM

Universidad Autónoma
del Estado de México

CENTRO UNIVERSITARIO UAEM TEXCOCO

“Aplicación de la Metodología Mobile-D™,
para la creación del prototipo de App
Caso de estudio: Draw-HánzÍ, sustentada en AR.”

ANEXO II

**Ejemplo de contrato comercial para
realización de un proyecto de desarrollo para
App.**



HUMANISMO
QUE TRANSFORMA

TEXCOCO, EDO. MÉX., OCTUBRE 2017

2. CONTRATO DE DESARROLLO DE SOFTWARE MÓVIL.

2.1. Ejemplo de contrato comercial para la realización de un proyecto de desarrollo de App (Solo sirve como guía para elaborar el propio).

En....., a.....de.....del.....
(la "fecha efectiva").

COMPARECEN

El presente contrato, por servicios de Desarrollo de Software Móvil, de una parte, "proveedor de servicios informáticos (nombre completo de la empresa)", a quien en lo sucesivo y para usos prácticos, se le denominara "únicamente las siglas del nombre de la empresa, en mayúsculas", representada legalmente por el(la) Sr.(a)..... "grado y nombre completo", quien da generales ser de nacionalidad....., mayor de edad, al corriente en sus obligaciones fiscales, con número de RFC..... con número de identificación vigente INE/Pasaporte/Licencia..... "carrera profesional", en libre ejercicio de su profesión, con cedula profesional No., quién desempeña actualmente el cargo de..... "cargo en la empresa", declarando como domicilio legal en "dirección completa del domicilio", a quien en lo sucesivo se le conocerá como el PROVEEDOR.

De otra parte..... " nombre de la empresa (cliente a quien en lo sucesivo y para usos prácticos, se le denominara "únicamente las siglas del nombre de la empresa, en mayúsculas", representada legalmente por el(la) Sr.(a)..... "grado y nombre completo", quien da generales ser de nacionalidad....., mayor de edad, al corriente en sus obligaciones fiscales, con número de RFC..... con número de identificación vigente INE/Pasaporte/Licencia..... "carrera profesional", en libre ejercicio de su profesión, con cedula profesional No., quién desempeña actualmente el cargo de..... "cargo en la empresa", declarando como domicilio legal en "dirección completa del domicilio", cuya sociedad

fue constituida ante el notario de....., el día....., inscrita en el registro mercantil de....., el día....., poder que exhibe en este acto y manifiesta vigente, a quien en lo sucesivo se le conocerá como el CLIENTE.

Ambas partes, para la ejecución de los mencionados Servicios de Desarrollo de Software Móvil y la producción de componentes descritos en el Cronograma A y Cronograma B, adjuntos en este documento e incorporados como referencia, acuerdan y de acuerdo con las leyes mexicanas y dentro del objeto social, se establece la posibilidad de celebrar un Contrato de Prestación de Servicios:

MANIFIESTAN

I. Que la empresa del CLIENTE es una entidad mercantil que tiene como objeto social.....

II. Que el PROVEEDOR Persona / Departamento, que representa a la empresa proveedora de servicios informáticos, tiene la experiencia necesaria para el desarrollo de software móvil.

III. Que ambas partes están interesadas en suscribir el presente contrato, con el objeto de que el PROVEEDOR, cree "*una aplicación informática móvil /software móvil / sistema móvil*", a medida de las necesidades del CLIENTE, cuya descripción general es la siguiente:.....

Por lo que, reconociéndose mutuamente la capacidad y la debida representación con que se actúan, ambas partes otorgan el presente CONTRATO DE DESARROLLO DE SOFTWARE MÓVIL, que se regirá por las siguiente:

CLÁUSULAS

PRIMERA. - Objeto del contrato

El presente contrato tiene como objeto de desarrollo, por parte del PROVEEDOR, la creación de un(as)....., descrito(tas) en el Anexo I de este contrato.

SEGUNDA. -Confidencialidad

Cada una de las partes se compromete a no difundir, bajo ninguno aspecto, la información científica o técnica perteneciente a la otra parte a la que haya podido tener acceso en el desarrollo del presente Contrato.

Los datos e informes obtenidos durante la realización del presente Contrato, así como los resultados finales, tendrán un carácter confidencial.

Ambas partes se comprometen a que todo el personal participante en el desarrollo del presente Contrato conozca y observe el compromiso de confidencialidad regulando por esta cláusula.

Las disposiciones de esta cláusula referentes a la confidencialidad subsistirán durante.....años después de la terminación del presente Contrato.

TERCERA. - Propiedad Intelectual

3.1. DEL CONTENIDO DEL CLIENTE. El contenido que el CLIENTE proporcioné para la realización del desarrollo, incluyendo todas las Marcas Registradas pre-existentes y material sujeto a derecho de autor, permanecerá bajo la propiedad exclusiva del CLIENTE, y el CLIENTE será el único titular de todos los derechos relacionados a éstos. El CLIENTE, por ende, concede al Proveedor una licencia no exclusiva, no transferible, para: utilizar, reproducir y modificar el contenido del CLIENTE, con el único propósito de la prestación de Servicios del PROVEEDOR y la producción de los componentes.

3.2. DE LOS MATERIALES DE TERCEROS. Todos los Materiales de Terceras Partes son de propiedad exclusiva de sus respectivos titulares. El PROVEEDOR deberá informar al CLIENTE de todos los Materiales de Terceros que puedan ser requeridos para ejecutar los Servicios del PROVEEDOR o bien para su integración en el Software Móvil Final. Bajo tales circunstancias, el PROVEEDOR deberá informar de cualquier necesidad de licenciamiento pertinente, cuyo costo se tendrá que prorratar (dividir) de la manera más pertinente para el beneficio de ambas partes.

3.3. DE LA ASIGNACIÓN DE DERECHOS DE AUTOR. Sujeto al cumplimiento de todos los pagos acordados en los Términos y Cronograma B de Pago, el PROVEEDOR otorgará al CLIENTE una licencia no exclusiva, paga, global y libre de regalías para instalar, utilizar y realizar copias del Software Móvil Final y toda documentación relacionada, conforme a los

términos y condiciones de este Acuerdo. El PROVEEDOR retendrá todo Derecho de Autor, patentes, secretos comerciales, y otros derechos de propiedad intelectual sobre el Software Móvil Final entregado. El CLIENTE se comprometerá a no remover, alterar, o encubrir cualquier notificación de Derecho de Autor, Marca Registrada, u otras notificaciones de derecho de propiedad que se incluyan en el Software Móvil Final.

3.3.1 DEL DERECHO A MODIFICACIONES. Adicionalmente, y también sujeto al cumplimiento de todos los pagos acordados en los Términos y Cronograma B de Pago, el CLIENTE podrá, a su discreción, realizar cambios o adiciones al Software Móvil Final creado bajo los términos y condiciones de este Acuerdo, pudiéndose contactar con otros proveedores para tales fines, sin incurrir en pagos al PROVEEDOR. El CLIENTE deberá pagar al PROVEEDOR un monto sujeto a negociación separada, en el caso de que solicite a éste la adición de nuevos Componentes y/o la modificación de Componentes Iniciales al Software Móvil Final ya entregado.

3.3.2. DE LA RESPONSABILIDAD DEL PROVEEDOR RESPECTO DEL DERECHO A MODIFICACIONES. El CLIENTE no podrá adjudicarle responsabilidad alguna al PROVEEDOR, en el caso de contactarse con terceras partes para realizar la adición de nuevos Componentes y/o la modificación de Componentes Iniciales / Finales al Software Móvil Final ya entregado, de producirse algún desperfecto, daño o falla en éste, a partir de la modificación del Software Móvil Final.

CUARTA. - Tarifas

En consideración de los Servicios que prestará el PROVEEDOR, el CLIENTE deberá abonar al PROVEEDOR tarifas en los montos y condiciones establecidos, en los Términos y Cronograma B de Pago, detallado y adjunto en el Anexo III, aquí para su consulta y respectiva referencia.

El costo total a pagar por parte del CLIENTE es deincrementados con el IVA correspondiente.

El pago total se realizará siguiendo los siguientes porcentajes de pago:

1. 30% del costo total al momento de la firma del presente contrato.

2. 60% con base al Cronograma B de pago, en sus Términos y condiciones correspondientes.
3. 10% en el momento en el que se lleva a cabo la instalación y entrega del manual de operación, necesaria para llevar a cabo su implementación

La emisión de la correspondiente FACTURA ELECTRÓNICA por parte del PROVEEDOR, será enviada vía correo electrónico dentro de un periodo de quince días naturales desde la recepción del último pago (inciso 3, anterior) expedido por el CLIENTE.

Para fines del párrafo anterior se establece por parte del CLIENTE, la siguiente dirección de correo electrónico.....y por parte del PROVEEDOR correspondiente dirección de correo electrónico.....

Ambas partes aceptan que, con base a los términos y condiciones establecidas y con base en los Artículos 2606 al 2615 del Código Civil Federal y conjuntamente con la Ley Reglamentaria en su Artículo 5°. Constitucional, también conocido como *Ley de profesiones*, en caso de incumplimiento en algunos de estos, se puede dar por terminado en buenos términos y sin previo aviso, el presente contrato por servicios de Desarrollo de Software Móvil.

- Si el CLIENTE es nacional: El abono de dichas cantidades será efectivo en la cuenta....., abierta en la entidad bancaria.....a nombre de.....
- Si el CLIENTE es extranjero: El abono de dichas cantidades será efectivo en la cuenta, abierta en la entidad bancariacon código de transferencia y referencia internacional

QUINTA. - Periodos generales del desarrollo

El desarrollo del software móvil / sistema móvil / aplicación informática móvil, objeto del presente contrato, cuenta con las siguientes fases:

- a) PERIODO INICIAL. Que comprende las fases de exploración y producción respectivamente, mismas que comprenden las tareas de Análisis y diseño del software móvil.

En la tarea de análisis

En la tarea de diseño, que comprende el diseño de las interfaces de usuario.....
En este periodo se procederá a la firma del presente Contrato y tendrá una duración dedías.

- b) PERIODO DE PRUEBAS. Que consistirá en la realización por parte del PROVEEDOR, de todas aquellas actividades necesarias para la comprobación del buen funcionamiento y usabilidad del software móvil desarrollado.

El periodo de pruebas tendrá una duración máxima de.....

- c) PERIODO DE MODIFICACIONES. Que podrá darse en caso de que, como consecuencia del periodo descrito anterior, surja la necesidad de llevar a cabo una serie de modificaciones, en software móvil desarrollado.

El periodo de modificaciones, siempre que sea necesaria, comenzará una vez terminada el periodo de pruebas, y tendrá una duración máxima de.....días.

- d) PERIODO DE ENTREGA E INSTALACIÓN definitiva del software móvil final. En este momento el PROVEEDOR hará entrega de toda la documentación de uso del programa informático, denominado el *manual de operaciones*.

- e) PERIODO DE CAPACITACIÓN. Que podrá impartir el PROVEEDOR, en caso de que el CLIENTE lo considere necesario, en las instalaciones que el CLIENTE establezca para dicho fin (cuyo costo corre por cuenta del mismo), pero que tendrá un costo negociable, independiente de este desarrollo. Con una duración máxima dehoras, a partir de la fecha, impartido por, de departamento, por parte del PROVEEDOR, a la siguiente lista de personas, del departamento Por parte del CLIENTE.

- f) PERIODO DE MANTENIMIENTO. Que será realizado por el PROVEEDOR, sin costo alguno con una duración máxima dedías, partiendo de la fecha de la aceptación y entrega por parte del CLIENTE, del software móvil final, misma aceptación será por escrito. En caso de que el CLIENTE considere necesario una

etapa de mantenimiento posterior, misma será con un costo negociable e independiente del presente desarrollo.

SEXTA. – Duración

La duración del presente Contrato será de días, contados a partir de la fecha de la firma del mismo, pudiendo prorrogarse por acuerdo escrito de ambas partes, si consideran oportuna la continuación para el buen fin del mismo.

SÉPTIMA. - Tiempos y Aprobación

7.1 TIEMPOS. El PROVEEDOR deberá priorizar la prestación de los servicios como sea necesario o acordado por ambas partes, y realizará esfuerzos comercialmente razonables para prestar los servicios. El CLIENTE acuerda revisar los componentes dentro del tiempo determinado para tales revisiones y se compromete:

a) *A aprobar y aceptar* los Componentes por *escrito* (convirtiendo a éstos en los componentes finales).

b) *Proveer comentarios y/o correcciones suficientes por escrito* para identificar las correcciones, objeciones y/o preocupaciones del CLIENTE al PROVEEDOR.

7.2 ACEPTACIÓN. El CLIENTE, dentro de los 2 días hábiles tras recibir cada componente, deberá notificar al PROVEEDOR, por escrito, de cualquier inexactitud del componente en cumplir las especificaciones acordadas por las partes, o de cualquier otra objeción, corrección, cambios o enmiendas que el CLIENTE desee realizar a tal Componente. Cualquier notificación por escrito deberá ser suficiente para identificar con claridad cualquier objeción, corrección o enmienda, y el PROVEEDOR deberá cumplir tales requerimientos en un tiempo sujeto al horario comercial de éste. Cualquier y toda objeción, corrección, cambio o enmienda estará sujeta a los términos y condiciones de este Contrato. En la ausencia de tal notificación por parte del CLIENTE, dentro del período de tiempo establecido, el Componente será considerado aceptado y, por ende, Final.

OCTAVA. – Lugar de presentación del servicio

El desarrollo del software móvil, objeto del presente contrato se efectuará en las instalaciones sede.....

En el caso del equipo desarrollador del PROVEEDOR, será representado por” cargo y nombre completo”, adscrito al departamento de, que tendrá como interlocutor válido por parte del CLIENTE a”cargo y nombre completo”, adscrito al departamento de

NOVENA. - Responsabilidades del cliente

El CLIENTE acepta ser responsable de cumplir lo siguiente en tiempos y formas razonables:

- a) Coordinación de cualquier toma de decisiones dentro de su propia de su Compañía / Empresa.
- b) Suministro de Contenido del CLIENTE en una forma adecuada para reproducción o incorporación en los Componentes sin preparación adicional.
- c) Revisión y aprobación Final, conforme a lo establecido en la sección 7 del presente Contrato.

DECIMA. - Reconocimiento y propiedad

Los derechos de explotación de la propiedad intelectual, incluyendo de manera expresa, el derecho de reproducción, transformación, distribución y comunicación pública, así como los derechos de propiedad industrial que, en su caso generen, corresponden de manera exclusiva al PROVEEDOR / CLIENTE.

El PROVEEDOR / CLIENTE podrá llevar a cabo, en su nombre, la protección y registro del software móvil desarrollado, ante las autoridades y/o instituciones pertinentes, bajo la legislación vigente y en caso de que procedimiento fuera posible, también podría llevar a cabo el registro de la modalidad de patente.

En todo caso, el titular de los derechos de explotación PROVEEDOR / CLIENTE deberá reconocer y respetar los derechos de propiedad inherentes a los autores del software móvil, en especial el derecho a ser designado como autor del mismo.

Dichos derechos protegen tanto al software móvil, que pueda resultar como a los datos, listados, diagramas, reportes y esquemas elaborados en la tarea de análisis, el manual de operación, los restantes datos y los materiales de apoyo, los símbolos de identificación, las reproducciones originales de los mismos, cualquier copia parcial o total realizada por la

propia empresa, los derechos legales de copia, las patentes, las marcas, los secretos comerciales y cualquier otro que pueda surgir en cumplimiento del presente Contrato.

El PROVEEDOR / CLIENTE ostentará la titularidad de dichos derechos durante toda la vida legal de los mismos y con un ámbito geográfico circunscritos a los países en los que se encuentre protegido.

UNDÉCIMA. – Información Confidencial

Cada una de las partes respectivamente acepta y se compromete, en conexión a este Contrato, a que puede recibir cierta información confidencial o propietaria, información o especificaciones técnicas igualmente propietarias, así como también materiales de la otra parte, incluyendo, pero no limitado a muestras parciales.

Cada una de las Partes, sus agentes y empleados deberán mantener en secreto toda información confidencial, se comprometerá a no difundir y no revelar bajo ningún aspecto, Información confidencial a ningún tercero y a no utilizar información confidencial alguna excepto en lo necesario para el cumplimiento de las obligaciones de desarrollo establecidas en este Contrato, salvo que fuere requerido por una Corte judicial o Autoridad Gubernamental. Sin perjuicio de lo anteriormente dicho, la información confidencial no deberá incluir ninguna información que esté en el dominio público o se dé a conocer públicamente sin error u omisión de la parte receptora, o bien es adecuadamente recibida por un tercero sin obligación de confidencialidad.

Los datos e informes obtenidos durante la realización del presente Contrato, así como los resultados finales, tendrán carácter confidencial.

Ambas partes, se comprometen a que todo el personal participante en el desarrollo del contrato conozca y observe el compromiso de confidencialidad regulado por esta cláusula. Las disposiciones de esta cláusula referentes a confidencialidad subsistirán durante.....años después de la terminación del presente Contrato.

DUODÉCIMA. – Relación de las partes

11.1 CONTRATISTA INDEPENDIENTE. El PROVEEDOR es un contratista independiente, no un empleado del CLIENTE o parte de la Compañía del CLIENTE o de cualquier entidad afiliada al CLIENTE.

El PROVEEDOR prestará sus Servicios bajo la dirección general del CLIENTE, pero el PROVEEDOR determinará la manera y los medios por los cuales se prestarán los Servicios. Este Contrato, no crea una sociedad u empresa conjunta, y ninguna de las partes está autorizada a actuar como representante de la otra parte respectivamente, salvo en las maneras establecidas por este Contrato. Todos los derechos, de existir alguno, concedidos al CLIENTE son contractuales en su naturaleza y definidos expresamente por el Contrato escrito entre ambas partes, en conjunto con los términos y condiciones de este Contrato.

11.2 NO EXCLUSIVIDAD. Las partes expresamente aceptan que este Contrato no crea una relación exclusiva entre las partes. El CLIENTE es libre de contactarse con otros proveedores, para solicitar servicios y/o asesoría, misma o de similar índole a aquellos prestados por el PROVEEDOR, y el PROVEEDOR tendrá el derecho de ofrecer y prestar servicios a otros, interactuar con otros clientes y publicitar los servicios prestados por el PROVEEDOR.

DECIMOTERCERA. - Garantías

13.1 Del CLIENTE. El CLIENTE garantiza al Proveedor que

- a) El CLIENTE es dueño de todo derecho, titularidad e interés, o bien tiene todo derecho y/o autoridad para permitir el uso del contenido del CLIENTE.
- b) Al sentido del CLIENTE, el contenido del CLIENTE no infringe los derechos de tercero alguno, y que el uso del contenido del CLIENTE, así como también de cualquier Marca Registrada vinculada al desarrollo del proyecto no infringe, ni infringirá los derechos de autor de tercero alguno.

13.2 Del PROVEEDOR.

- a) El PROVEEDOR garantiza al CLIENTE que el PROVEEDOR prestará los servicios del PROVEEDOR y producirá los componentes identificados en este Contrato de manera profesional y responsable, conforme a todos los estándares profesionales relevantes a tales servicios.
- b) El PROVEEDOR garantiza que:
 - l) Los componentes y/o productos finales, serán trabajo original del PROVEEDOR.

II) Al entendimiento del PROVEEDOR, el Software Móvil Final entregado por el PROVEEDOR, no infringe los derechos de tercero alguno, y el uso del mismo en conexión con el desarrollo del proyecto no infringirá los derechos de tercero alguno.

c) El PROVEEDOR garantiza que:

Se establece un plazo de garantía de máxima cobertura los primerosdías naturales, a partir de la fecha de la instalación definitiva del software móvil en los dispositivos y/o equipos del CLIENTE. Si el CLIENTE no identifica la existencia de errores y/o defectos a el PROVEEDOR durante el tiempo referido, se considerará que está conforme en todos los aspectos con el funcionamiento del Software Móvil Final, renunciando, a partir de entonces, a cualquier reclamación.

Esta garantía será válida siempre y cuando el CLIENTE utilice el Software Móvil Final desarrollado, conforme a las especificaciones técnicas realizadas por el equipo de desarrollo del PROVEEDOR.

DECIMOCUARTA. – Responsabilidades

El PROVEEDOR no será responsable de las pérdidas o daños sufridos por el CLIENTE, sus agentes, empleados o clientes, directa e indirectamente originados por los errores del Software Móvil Final, dentro de sus operaciones.

DECIMOQUINTA. - Actividades Adicionales

Cualquier actividad adicional relacionada con el software móvil desarrollado, durante el propio proceso de desarrollo, tales como mantenimiento, ampliaciones y/o modificaciones, quedaran fijadas por escrito en un nuevo Contrato y/o Acuerdo posterior al presente Contrato.

DECIMOSEXTA. -Modificación, vigencia y terminación del presente Contrato.

Ambas partes participantes, podrán modificar el presente Contrato, en cualquier momento por mutuo acuerdo y por escrito, dentro del periodo de vigencia del mismo.

Este Contrato comenzará a partir de la fecha efectiva y permanecerá vigente hasta que los servicios estén completados y los componentes finales y el Software Móvil Final hayan sido

entregados, con las debidas conformidades y aprobaciones por escrito del CLIENTE, producidas al cierre de cada componente entregado.

El presente contrato podrá ser diluido/rescindido en cualquier momento por las siguientes causas:

1. POR MUTUO ACUERDO DE LAS PARTES.

De cualquiera de las partes, de manera efectiva e inmediata apenas recibida la notificación de mutuo acuerdo de las partes, o si alguna de las partes:

a) Se encuentra insolvente o se declara en bancarrota, si este fuese el motivo de alguna de las partes y se ve obligada a disolver este contrato por dicha causa, deberá comunicarlo de forma fehaciente a la otra parte de inmediato.

b) Quebranta cualquiera de sus responsabilidades u obligaciones materiales en el marco de este Contrato, notificándosele, mediante el método de comunicación fehaciente, e indicara de manera clara las causas que originan dicho incumplimiento, cuyo quebrantamiento no haya sido remediado dentro los 10 días de recibida la notificación sobre tal incumplimiento.

En caso de no subsanarse por la parte incumplidora, la otra parte podrá disolver el presente Contrato y se valorará el trabajo realizado hasta el momento, los gastos comprendidos y los daños y perjuicios causados por dicho incumplimiento. Esta valoración nunca superará el valor económico total del presente Contrato.

2. AL EXPIRAR O RESCINDIRSE ESTE ACUERDO:

a) Cada una de las partes deberá devolver o, a petición de la parte correspondiente, destruir la Información confidencial de la otra parte.

b) Salvo establecido contrariamente por el presente Contrato, todos los derechos y obligaciones de cada parte, referidos exclusivamente a los servicios, no perecerán.

DECIMOSÉPTIMA. -Indemnización

17.1 POR PARTE DEL CLIENTE. El CLIENTE acuerda indemnizar, proteger y resguardar al PROVEEDOR de cualquier y todo daño, responsabilidad, costo, pérdida o gasto derivado de cualquier reclamo, demanda o acción judicial iniciada por un tercero, en consecuencia,

del quebrantamiento de cualquiera de las responsabilidades u obligaciones, y garantías del CLIENTE, debidamente establecidas en el presente Contrato.

17.2 POR PARTE DEL PROVEEDOR. Sujeto a los términos, condiciones y expresas garantías declaradas en el presente Contrato, el PROVEEDOR acepta indemnizar, proteger y resguardar al CLIENTE de cualquier y todo daño, responsabilidad, costo, pérdida o gasto incurrido por cualquier hecho inconsistente con las responsabilidades y garantías que el PROVEEDOR aceptó debidamente en el presente Contrato, salvo aquellos reclamos, daños, responsabilidades, costos, pérdidas o gastos directamente derivados de la negligencia o falta de conducta por parte del CLIENTE.

DECIMOCTAVA. - DISPOSICIONES GENERALES

18.1 NOTIFICACIONES. Todas las notificaciones a entregarse durante la vigencia del presente Contrato serán transmitidas por escrito vía correo electrónico y/o alternativas similares, con acuse de recibo, y deberán ser enviadas a las direcciones de correo electrónico consignadas en el Anexo IV a la firma del presente Contrato, salvo que se dé una notificación de cambio de dirección de correo electrónico por escrito. La notificación será efectiva apenas sea recibida o cualquiera de las partes reciba el correspondiente acuse de recibo.

18.2 JURISDICCIÓN. La formación, construcción, ejecución y vigencia del presente Contrato, estarán en concordancia con las leyes y normas de los Estado Unidos Mexicanos y del Estado de México, sin consideración respecto al conflicto de disposiciones legales de cualquier otra jurisdicción.

18.3 DISOLUCIÓN. Siempre que sea posible, cada disposición del presente Contrato será interpretada de tal manera que sea efectiva y válida bajo la ley aplicable, pero si por alguna disposición del presente Contrato se considera inválida o inefectiva, el resto del Contrato se mantendrá en plena vigencia y la disposición inválida deberá ser respaldada por otra debidamente válida.

18.4 TÍTULOS. La numeración y título de las diferentes secciones de este Contrato sirven al único fin de comodidad y referencia y no afectarán al alcance, significado, intención o interpretación de las disposiciones de este Contrato, ni tales títulos tendrán efecto legal alguno.

18.5 INTEGRACIÓN. El presente Contrato comprende el total entendimientos de las partes respecto al tópico en cuestión, y sustituye y combina todos los previos acuerdos, entendimientos y discusiones entre las partes, referidos al tópico que concierne a este Contrato.

Con su firma, las Partes presentes acuerdan aceptar todos los términos y condiciones de este Acuerdo, efectivo a partir de la última fecha de firma, y cada signatario declara poseer total autoridad para tomar parte en este Acuerdo y vincular su respectiva Parte a todos los términos y condiciones aquí establecidos.

DECIMONOVENA. - Definiciones

Empleadas en la totalidad de este Contrato:

19.1 “*Contrato*” significa la totalidad de este documento, los documentos de la Propuesta (si los hubiere), el Cronograma A, el Cronograma B, en conjunto con cualquier otro Suplemento, Anexo, o Cronogramas adicionales que se incorporen como referencias aquí.

19.2 “*Contenido del Cliente*” significa todos los materiales, información, fotografías, escritos y otros contenidos provistos por el Cliente para el uso en la preparación y/o incorporación en los componentes.

19.3 “*Componentes*” significa los servicios y el producto, mutuamente acordados tanto por el CLIENTE como el PROVEEDOR, que el PROVEEDOR entregará al CLIENTE, en la forma y medios que especifica el Cronograma A.

19.4 “*Herramientas de Desarrollo*” significa todas las herramientas de desarrollo creadas y/o utilizadas por el PROVEEDOR para la prestación de los servicios, incluyendo, pero no limitado a: software pre-existente y creado con su correspondiente código fuente, herramientas de desarrollo Web, tipografías, y herramientas de aplicación, en conjunto con cualquier otro software, u otras invenciones (patentables o no patentables) y conceptos generales no alcanzados por el derecho de autor, como los elementos de arquitectura, disposición, navegación (Web, si correspondiera) y funcionalidad de los Componentes.

19.5 “*Software Móvil Final*” significa todo el contenido desarrollado o creado por el PROVEEDOR de manera exclusiva e incorporado al proyecto como parte de los componentes finales, incluyendo, pero no limitándose a elementos visuales, interfaces, herramientas lógicas, datos, y código fuente, adaptaciones y/o modificaciones al contenido

del CLIENTE, y la selección, arreglo y coordinación de dichos elementos por parte del PROVEEDOR, en conjunto con materiales de terceros, si los hubiere, aprobados y aceptados debidamente por el CLIENTE.

19.6 “*Componentes Finales*” significa las versiones finales de los componentes y/o productos, entregadas por el PROVEEDOR y aprobadas y aceptadas por el CLIENTE.

19.7 “*Muestras Preliminares*” significa todo conjunto de sin limitarse a: conceptos, previsualizaciones de funcionalidad, presentaciones u otros documentos y especificaciones preliminares, desarrolladas por el PROVEEDOR y que pueden o no ser mostradas y/o entregadas al CLIENTE para su consideración.

19.8 “*Proyecto*” significa el alcance y propósito del uso identificado del producto (software móvil) por parte del CLIENTE.

19.9 “*Servicios del PROVEEDOR*” significa todos los servicios y el resultado del trabajo que el PROVEEDOR entregará al CLIENTE como está descrito y definido en cada componente.

19.10 “*Materiales de Terceros*” significa componentes propietarios de terceras partes que se incorporan a los componentes finales, incluyendo (pero no limitado a): íconos, tipografías, código fuente, librerías de software o componentes relevantes a la implementación de los componentes finales.

19.11 “*Marcas Registradas*” significa nombres comerciales, palabras, símbolos, logotipos u otros dispositivos y/o diseños utilizados para designar el origen de bienes o servicios.

VIGÉSIMO. – Anexos

Anexo I. REQUISITOS: Documento que identifica y especifica los requerimientos iniciales del Software Móvil Final.

Anexo II. CRONOGRAMA A: Documento que describe fielmente cada uno de los componentes que serán creados por el PROVEEDOR al CLIENTE al término del presente Contrato.

Anexo III. CRONOGRAMA B: Documento que contiene el debido presupuesto y que especifica los términos de pagos, conjuntamente especificado con un cronograma de pago.

Anexo IV. NOTIFICACIONES Y PERSONAL PARTICIPANTE: Documento que especifica por escrito, los nombres completos, números telefónicos, correos electrónicos, números

celulares y en caso de considerarlo necesario, redes sociales, para contactar a cada uno de los integrantes, de ambas partes respectivamente.

Anexo V. DESCRIPCIÓN DEL TRABAJO. Documento que describirá brevemente el objeto de desarrollo del presente Contrato en términos generales, a manera de que sea comprendido en su totalidad.

POR EL “PROVEEDOR”

POR EL “CLIENTE”

GRADO Y NOMBRE COMPLETO

GRADO Y NOMBRE COMPLETO

TESTIGOS

POR EL “PROVEEDOR”

POR EL “CLIENTE”

GRADO Y NOMBRE COMPLETO

GRADO Y NOMBRE COMPLETO



UAEM

Universidad Autónoma
del Estado de México

CENTRO UNIVERSITARIO UAEM TEXCOCO

“Aplicación de la Metodología Mobile-D™,
para la creación del prototipo de App
Caso de estudio: Draw-HánzÍ, sustentada en AR.”

ANEXO III

Manual de desarrollo de Apps en México.



HUMANISMO
QUE TRANSFORMA

TEXCOCO, EDO. MÉX., OCTUBRE 2017

3. MANUAL DE DESARROLLO DE APPS EN MÉXICO.

3.1. Manual estandarizado sobre el proceso de creación de una App, basado en la metodología Mobile-D™.

Un MANUAL es un documento que contiene entre muchas otras cosas, una forma ordenada y sistemática de información (instrucciones), sobre los procedimientos a seguir por un desarrollador Freelance o como parte de una Startup y/o PyME IT Mexicana, necesarios para la mejor ejecución de su trabajo.

Un MANUAL DE PROCEDIMIENTOS es útil porque, permite conocer el funcionamiento interno de las dependencias en lo que se refiere a la descripción de tareas, ubicación, requerimientos, y a los puestos responsables de su ejecución, mostrando su función, sus soportes documentales y la autorización que se requiere.

El objetivo fundamental de un manual, independientemente de su tipo, es explicar en términos accesibles y prácticos, el ¿Por qué?, de las decisiones; y el ¿Cómo?, se debe aplicar en la práctica.

Con base, en lo anterior y a la experiencia en el uso de la ya mencionada Metodología Mobile-D™, se ha formulado una definición propia, siendo la siguiente:

“Los MANUALES DE PROCEDIMIENTOS, también conocidos bajo el nombre de manuales de operación, manuales de prácticas estandarizadas, manuales de introducción sobre un trabajo especializado, manual de tramites o manual de métodos de trabajo; contienen la información sobre el conjunto de operaciones, acciones y/o etapas, en forma cronológica establecidos para llevar a cabo un determinado tipo de trabajo específico, presentando una visión en conjunto, sirviendo como medio de integración y orientación, hacia el personal de nuevo ingreso o novato, en la tarea, dentro de una organización y que facilite su correcta incorporación a las distintas funciones que desarrollara en lo sucesivo”.

La función del “manual de desarrollo de Apps en México”, generado del presente trabajo de investigación científica tecnológica de tesis, es el promover la creación, difusión e implementación de una guía de mejores prácticas para el desarrollo de Apps de alta calidad, sustentadas en las directrices de la metodología Mobile-D™, con el siguiente propósito:

- Permitir un ahorro de tiempo, esfuerzo e inversión económica, en la integración y orientación en la descripción de tareas, de requerimientos, de ubicación hacia los puestos de trabajo responsables de su ejecución.

3.1.1. Manual estandarizado sobre el proceso de creación de una App, basado en la metodología Mobile-DTM, ilustraciones.

NOTA 1: El manual ha sido creado como formato de "infografía", por lo que algunos bloques no estarán totalmente visibles en este formato; se aconseja utilizar el zoom necesario en dichas imágenes y/o un programa especializado que permita su correcta visualización.

Figura MA001. Módulo #01 título de bienvenida.



Fuente: Elaboración propia, 2017.

Figura MA002. Módulo #02 resumen del objetivo del manual.



Fuente: Elaboración propia, 2017.

Figura MA003. Módulo #03 descripción de los primeros pasos.

#1 Primeros Pasos

Tal vez acabas de graduarte de una carrera profesional afín a la tecnología o estás en busca de obtener experiencia en el desarrollo de Apps, por la satisfacción de emplearte en algo de tu agrado; independientemente de cuál sea la razón que te motiva, necesitarás conseguir un grupo de clientes y/o inversores potenciales rápidamente, situación que puede llegar a parecer una tarea extremadamente abrumadora.

Sin embargo, el presente manual de desarrollo de Apps en México, ha sido generado con la intención de ofrecerte un contexto en el cual puedas pertenecer organizado y enfocado en el objetivo de buscar oportunidades de desarrollo de Apps, adaptar tus recursos, así como, destacar durante el proceso de producción al generar Apps con alta calidad, en el menor tiempo posible, aumentando las posibilidades de tener éxito y adquirir mayor experiencia en el ramo del desarrollo de productos de software móvil en México.

Todos nosotros en algún momento de nuestras vidas, así mismo, como en un período entusiasta de la tecnología y su desarrollo, nos hemos enfrentado a las siguientes observaciones: ¡Quiero crear una App, ¡Hoy he tenido la idea más creativa para la App perfecta!, ¡esta App no la he pensado nadie antes y estoy seguro! que será muy popular!, ¡he encontrado la manera de resolver un problema en determinado lugar, con el uso de esta App! O ¡Estoy seguro! que esta idea de App, tendrá mejor aceptación!.

Tu proyección resultante a estas afirmaciones determinará la generación de un producto de software móvil "App", el incógnito más frecuente a que todos nos enfrentamos con o sin experiencia previa, es saber por dónde empezar, nadie nos explica paso a paso como empezar mi primera y/o una App de alta calidad y éxito, de manera fácil.

Para que una empresa, institución, organización, comercio y/o inversor te seleccione como proveedor de servicios tecnológicos o decida invertir en ti, como desarrollador de software, es decir, creador de Apps, ya sea de forma Freelance, como integrante de una área específica o como parte de una Startup y/o PyME IT Mexicana, independientemente del respaldo en conocimiento, capacidad y habilidad de desarrollo con el que cuentes, es igualmente de importante, que realices un esfuerzo justo en la elaboración de una propuesta de venta exitosa.

En síntesis y con la finalidad de encaminar tus primeros pasos, te ofrecemos la siguiente serie de tips:

FORMULA LA PROPUESTA A MEDIDA DE LA NECESIDAD DEL CLIENTE Y/O INVERSOR

El momento antes de iniciar un proceso de venta, analizar, investigar y buscar todo sobre el cliente, para entender la necesidad que se tiene dentro de la empresa (short speech) inicial puede asegurar el concepto en una cita, siempre y cuando el cliente necesite o el dolor (pain) tiene, cuánto espera invertir y de que forma cree que es la más correcta, en que lo puedes ayudar.

ORGANIZA CENAS O COMIDAS FORMALES O SEMI-FORMALES E ESTRATEGICAS.

Según Jorge Sánchez, productor ejecutivo de ProdigiousTV "La clave del éxito de una cena o comida y/o reuniones no está en el costo del menú sino en la atención a los detalles que se tuvo para planear el evento, el lugar que eliges hablar de lo que tanto mundo tienes, si eres de buen gusto, moderno o clásico y si eres alguien que se preocupa por los detalles y es consciente con las necesidades de su invitado".

ENTRA POR LA PUERTA GRANDE

Busca llegar lo más alto que puedas. Apóyate en redes sociales como LinkedIn para buscar los nombres, conexiones y estadísticas de clientes potenciales. Con esa información podrás prepararte e iniciar una cita o presentación en persona, si el cliente regional las prefiere, ya presentada más difícil. Tienen cuenta que no siempre se vende, lo importante es conectar y dejar la puerta abierta para futuras negociaciones.

TEN PERFECTAMENTE CLARO CUÁL ES EL VALOR AGREGADO DE TU PRODUCTO DE SOFTWARE MÓVIL

¿Cómo tu App puede generar valor al cliente estratégico y/o inversor potencial a cubrir su dolor o necesidad?

DESTACA TUS ÉXITOS PREVIOS Y TU CREDIBILIDAD

Apoya tu propuesta con ejemplos que muestren cómo creaste un tipo bien en un momento de negocio rentable, cómo dirigiste el equipo o llevaste a cabo las medidas necesarias de esta manera demostrar de cierta veracidad en las negociaciones sucesivas y se mantendrá en tus probabilidades de éxito y/o contratación.

SE CONSCIENTE DE LO QUE PIDES Y DE ¿POR QUÉ?

Los mejores desarrolladores nunca se dejan llevar solo por las ideas. Al final, las soluciones que otros no deberían estar involucradas a un aspecto necesariamente empresarial. En otras palabras, lo mejor es comprar las cosas que el proveedor que trabaja del negocio con tu propio presupuesto. Una necesidad inmediata o si es posible, y que empiezo con la filosofía general de la participación y de todos sus miembros.

ANALIZA CONTRA QUIÉNES ESTÁS COMPTIENDO

Aun si no has una conexión directa, es posible que el proyecto compita contra otros. Trata de investigar cómo argumentar tu idea o propuesta en funciones y módulos manejables, con el fin de que toda persona que trabaje en el proyecto se beneficie de ella y no se pierda el interés. Al final, el cliente desarrollador en un equipo de desarrollo, es que la presentación y que se nota. Puedes aprovechar este aspecto para hacer mención a tu trayectoria empresarial y/o mostrar a algún colega que consideres competente para crear un producto rentable de esta manera se muestra un ejemplo.

ELABORA UN CONTRATO QUE INCLUYA TODA LA INFORMACIÓN RELEVANTE

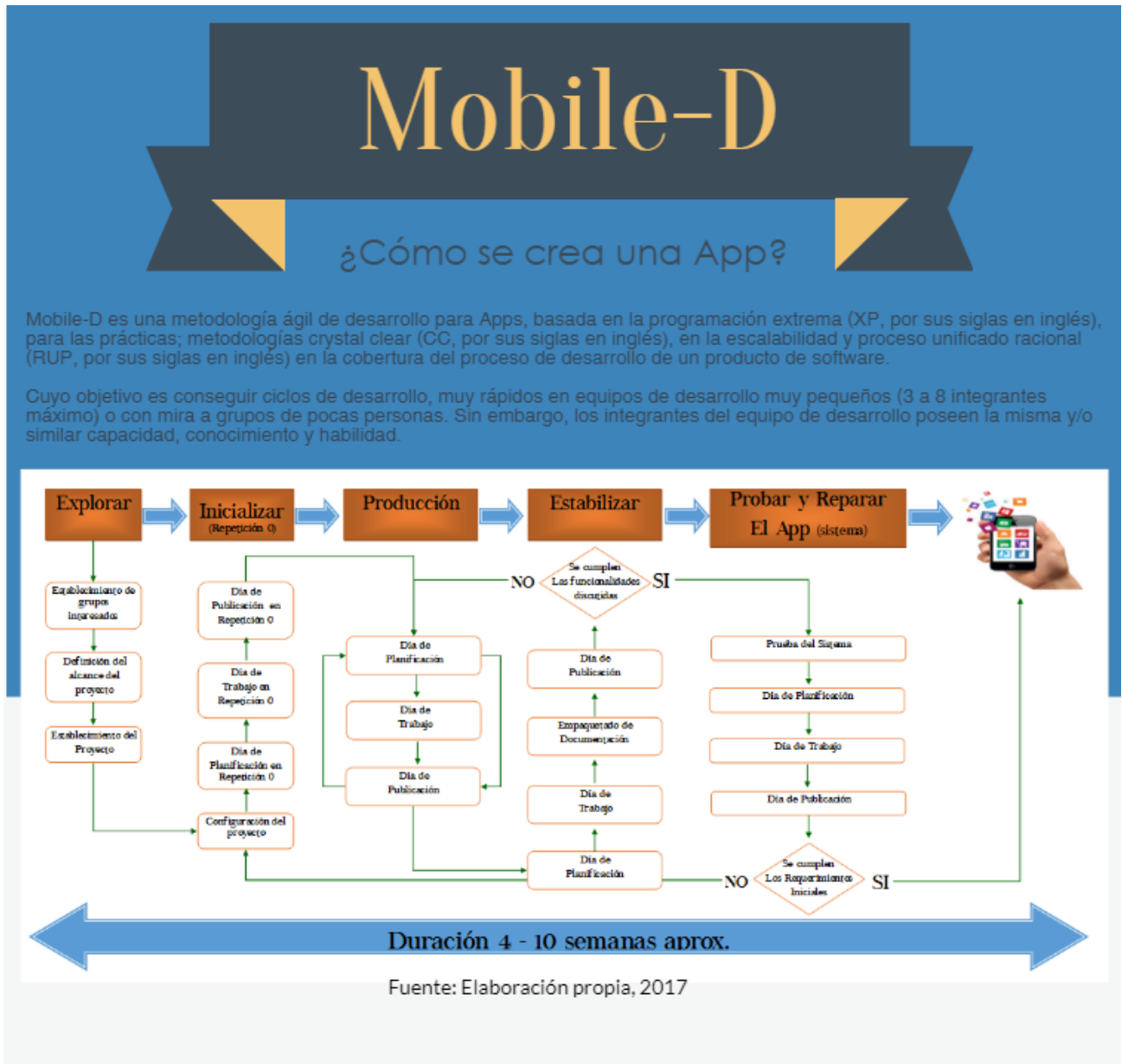
La propuesta debe de ir acompañada de un buen contrato que contenga, todo lo solicitado por el cliente y/o inversor potencial (o. e. precio, plazos, aspectos legales, por mencionar algunos), así como otros datos básicos que permitan regular cualquier aspecto de forma correcta, sea mediante la firma de declaración o simplemente. (Verse ANEXO II)

PRESENTA TU PROPUESTA / CONTRATO EN LOS TIEMPOS Y FORMA ADECUADOS

Considera no solo todo lo anterior, sino también la manera de crear un impacto que la diferencia de la competencia. Presentar tu propuesta / contrato primero muestra atención por tu trabajo, acompañar la presentación con un bar de reuniones (una de aspecto formal y otra de aspecto más familiar), pueden ser factores y/o detalles clave para ganar el proyecto, lograr la venta o inversión y/o contratación, al mostrar una forma precisa y eficiente de trabajar.

Fuente: Elaboración propia, 2017.

Figura MA004. Módulo #04 descripción inicial de la Metodología Mobile-D™.



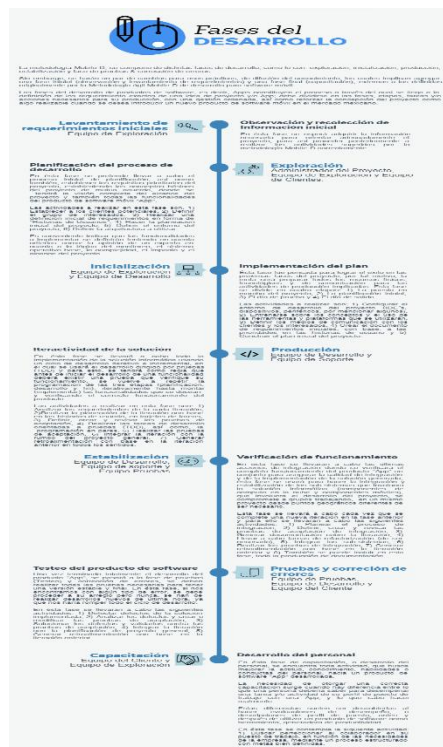
Fuente: Elaboración propia, 2017.

Figura MA005. Módulo #05 descripción del equipo de trabajo según, la Metodología Mobile-D™.



Fuente: Elaboración propia, 2017.

Figura MA006. Módulo #06 descripción de las fases de la Metodología Mobile-D™.



Fuente: Elaboración propia, 2017.

Figura MA007. Módulo #07 definición del termino aplicación móvil.



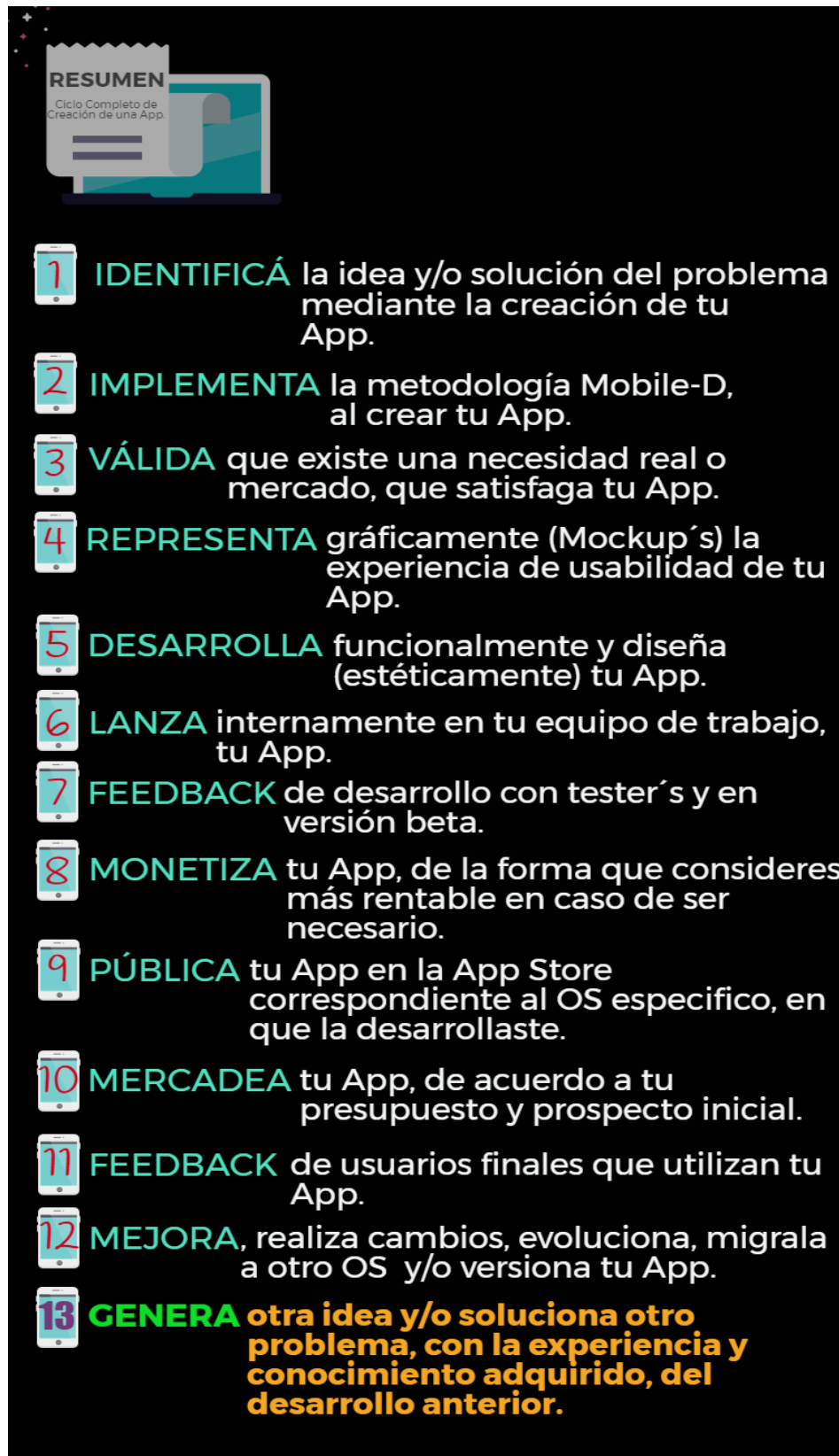
Fuente: Elaboración propia, 2017.

Figura MA008. Módulo #08 aspectos a considerar al validar una idea de App.



Fuente: Elaboración propia, 2017.

Figura MA010. Módulo #10 Resumen del ciclo de desarrollo de una App.



Fuente: Elaboración propia, 2017.

Figura MA011. Módulo #11 aviso legal sobre el manual.



The slide features a blue background with a central logo consisting of a green padlock inside a circular arrow. To the right of the logo, the text "Aviso Legal" is written in a bold, green font. Below the logo and title, there are five paragraphs of white text providing legal disclaimers and information about the Mobile-D methodology. At the bottom, a small note in white text states: "NOTA: Las imágenes e iconos son propiedad de Piktochart & Freepik respectivamente."

Aviso Legal

Mobile-D es una metodología registrada por el VTT Electronics (Centro de investigación Técnica de Finlandia Ltd).

Todas las demás marcas registradas que se mencionan, usan y/o citan en el presente manual son propiedad de los respectivos titulares.

El presente manual cita estas marcas por la importancia considerable, que proporcionan referentes en los temas que se tratan, buscando únicamente fines puramente divulgativos. En ningún momento se busca con su mención, el uso interesado de estas marcas, ni manifestar cualquier participación y/o autoría de las mismas.

Nada de lo contenido en el presente manual debe ser entendido como concesión, por implicación o de otra forma, y cualquier licencia o derecho para las marcas registradas deben tener la autorización escrita de los terceros propietarios de la marca.

Por otro lado, los autores del presente manual renuncian expresamente a asumir cualquier responsabilidad relacionada con la publicación de las marcas registradas, en este documento en cuanto al uso de ninguna en particular y se exime de la responsabilidad de la utilización de dichas marcas por terceros.

El carácter del presente manual es únicamente de guía estandarizada formativa, buscando en todo momento facilitar a los lectores la comprensión, adaptación y divulgación de las disciplinas, técnicas, prácticas, metodologías, estándares y normas presentes en el ámbito del desarrollo de software móvil de alta calidad a nivel mundial.

NOTA: Las imágenes e iconos son propiedad de Piktochart & Freepik respectivamente.

Fuente: Elaboración propia, 2017.

Figura MA012. Módulo #012 información de contacto.



The slide has a dark blue background with white text. At the top, it says "Creado por: Trejo Quintero Sandy & Moreno Gorgonio Ulises." followed by "CU UAEM TEXCOCO, Licenciatura en Informática Administrativa 2017." Below this, there is a Facebook icon and the text "Methodoogy Mobile-D en México". At the bottom, there are two email addresses: "mobile.d.commx@gmail.com" and "commx.mobile.d@gmail.com", with a small envelope icon between them.

**Creado por: Trejo Quintero Sandy
&
Moreno Gorgonio Ulises.**

CU UAEM TEXCOCO, Licenciatura en Informática Administrativa 2017.

f Methodoogy Mobile-D en México

mobile.d.commx@gmail.com  commx.mobile.d@gmail.com

Fuente: Elaboración propia, 2017.





UAEM

Universidad Autónoma
del Estado de México

CENTRO UNIVERSITARIO UAEM TEXCOCO

“Aplicación de la Metodología Mobile-D™,
para la creación del prototipo de App
Caso de estudio: Draw-HánzÍ, sustentada en AR.”

ANEXO IV

Manual de ejecución y funcionalidad del prototipo de App.
Caso de Estudio: DRAW- HÁNZÍ.



HUMANISMO
QUE TRANSFORMA

TEXCOCO, EDO. MÉX., OCTUBRE 2017

4. MANUAL DE OPERACIÓN DE LA APP DRAW- HÁNZÍ.

4.1. Requisitos mínimos para la excelente operación de la App

Antes de poder instalar la aplicación Draw Hanzi primero tendremos que ver si nuestro dispositivo cubre con los requisitos mínimos para que la app pueda funcionar de manera correcta.

4.1.1 Requisitos del sistema

- Sistema Operativo: Android KitKat 4.4 o superior.
- Memoria RAM: 2GB.
- Memoria ROM: 16GB.

Tabla MAED01. Versiones de Android Recomendadas.

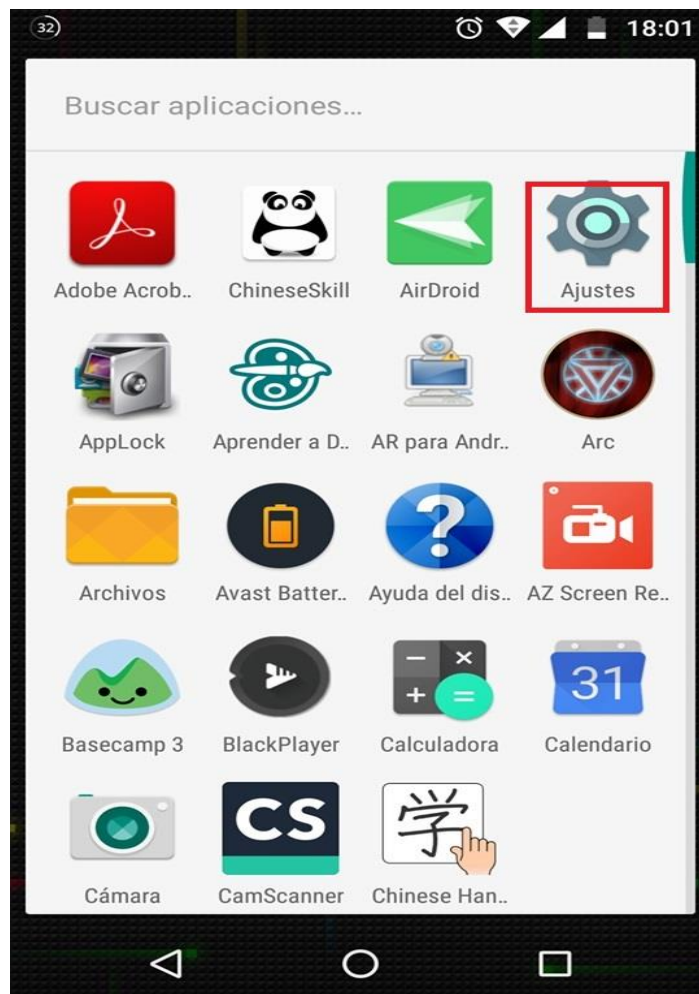
Versiones de OS Android recomendadas para la instalación de Draw-HánzÍ			
Nombre código	Número de versión	Fecha de lanzamiento	Nivel de API
KitKat	4.4 – 4.4.4, 4.4W–4.4W.2	31 de octubre de 2013	19–20
Lollipop	5.0–5.1.1	12 de noviembre de 2014	21–22
Marshmallow	6.0–6.0.1	5 de octubre de 2015	23
Nougat	7.0 - 7.1.2	22 de agosto de 2016	24 - 25

Fuente: Elaboración propia, 2017.

4.1.2 Configuración para correcta instalación

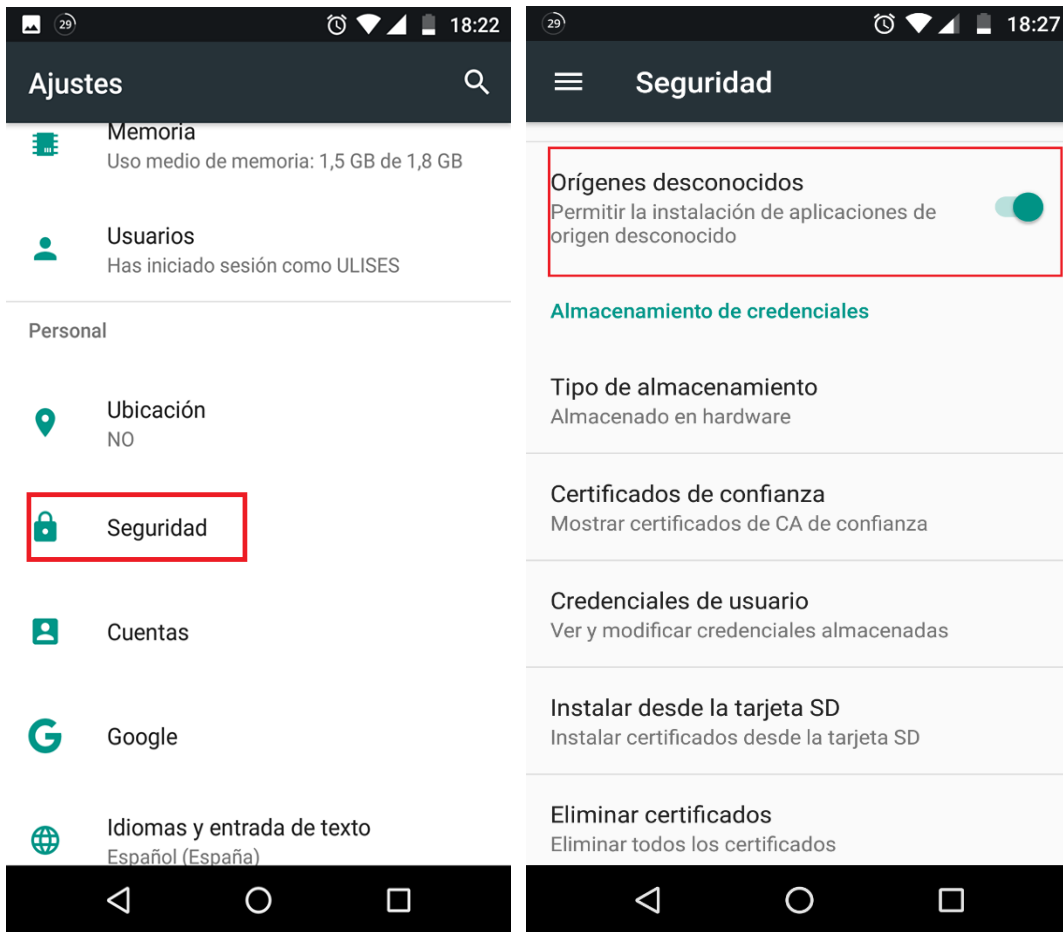
Para poder instalar correctamente, el prototipo de App Draw-Hánzǐ, es necesario hacer algunas configuraciones a nuestro dispositivo móvil (Smartphone, Tablet, por mencionar algunos), por lo que tendremos que abrir la opción ajustes; una vez dentro de la opción ajustes buscamos la opción Seguridad y la seleccionamos después buscaremos la opción Orígenes desconocidos y la activamos, esta opción nos permitirá instalar el .Apk y seguir los pasos como se ilustran en las siguientes figuras MAEDH01 y MAEDH02 respectivamente:

Figura MAEDH01. Selección de la opción Ajustes



Fuente: Elaboración propia, 2017.

Figura MAEDH02. Configuración para la correcta instalación de archivos. APK

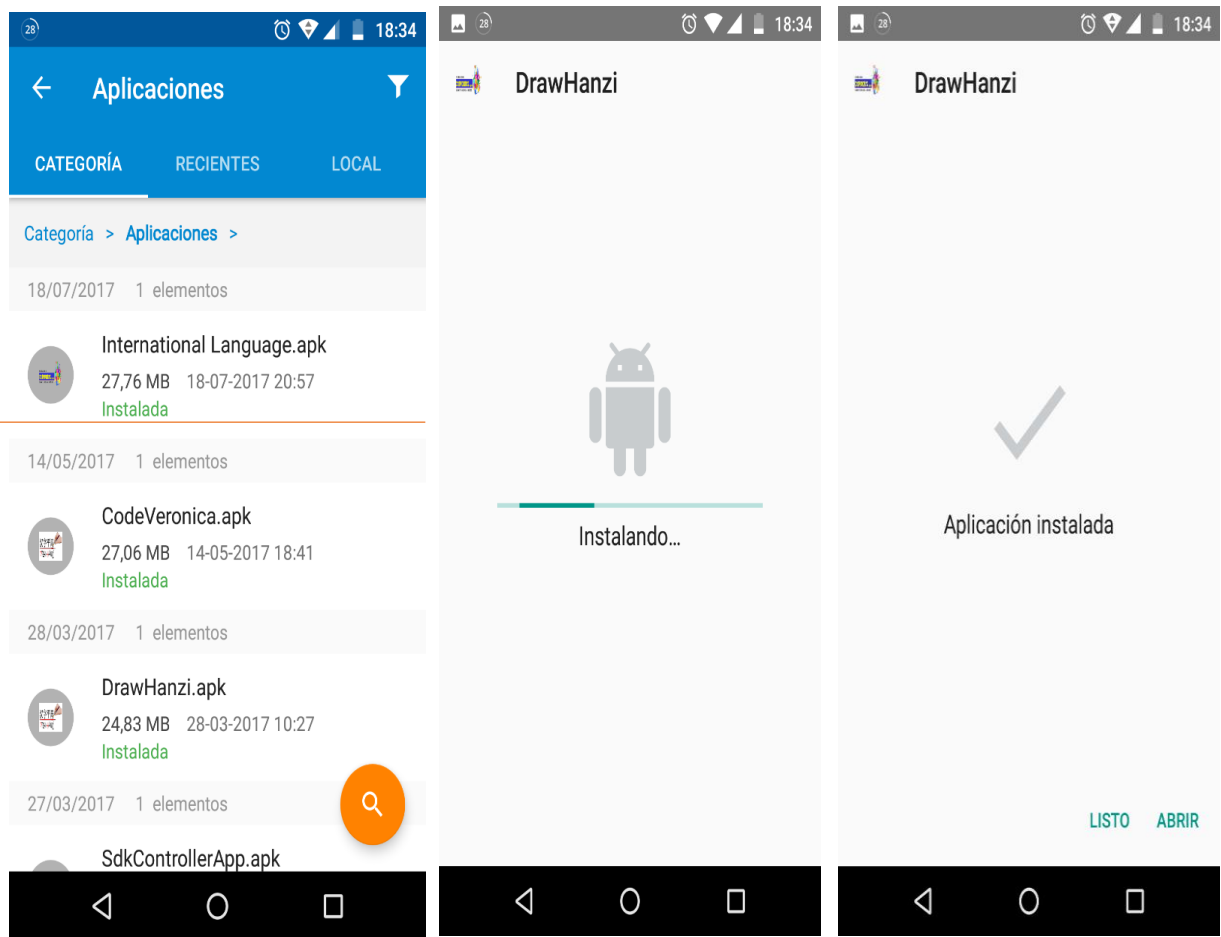


Fuente: Elaboración propia, 2017.

4.1.3 Instalación del .Apk del prototipo de App Draw- Hánzí.

Una vez hechas las configuraciones anteriores, de forma correcta, se procederá a instalar el .Apk de la App Draw- Hánzí.

Figura MAEDH03. Ilustración de instalación de App Draw-Hánzí

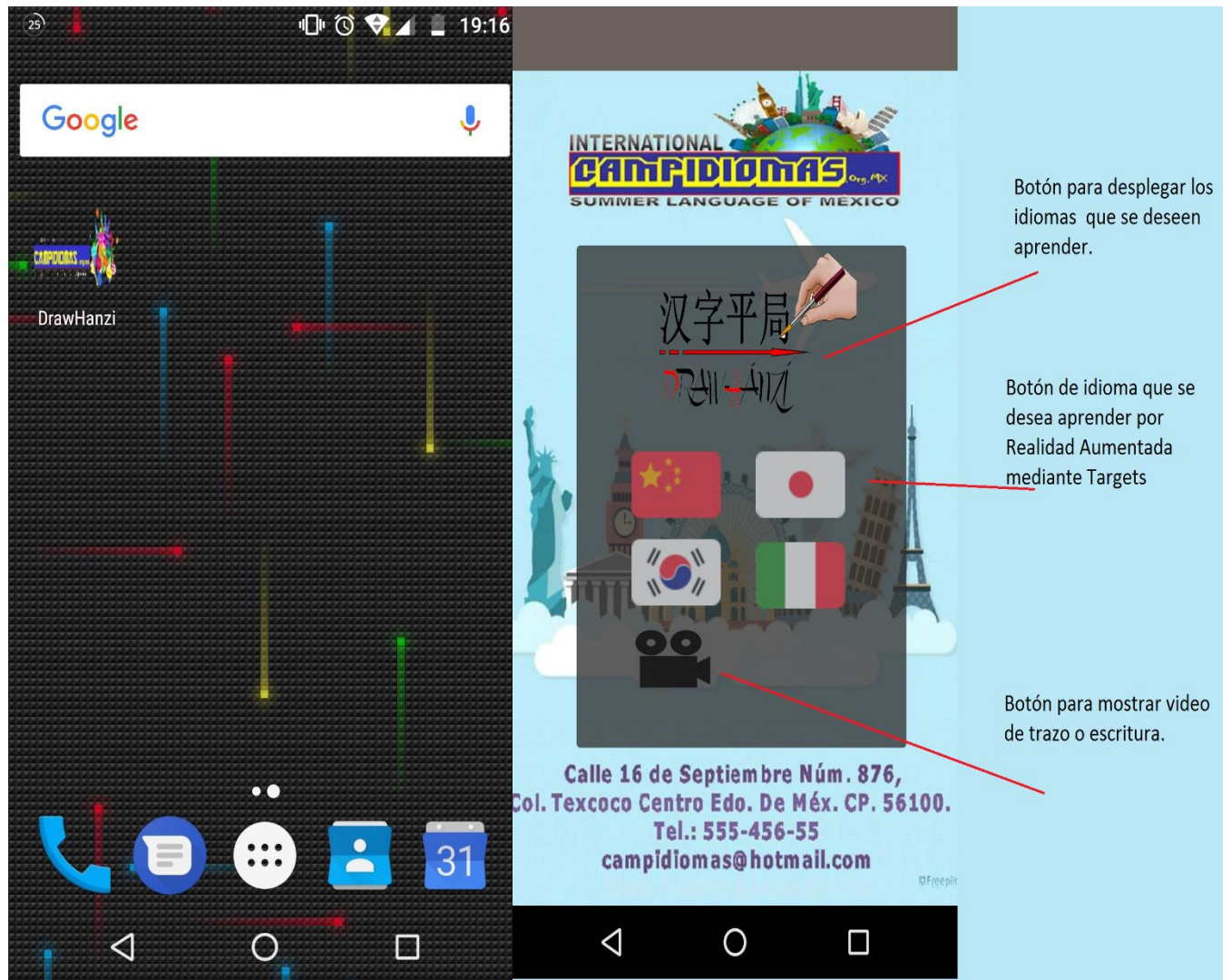


Fuente: Elaboración propia, 2017.

4.1.4 Apertura de usabilidad de la App Draw-Hánzí

Una vez instalado el archivo .Apk correctamente como se ilustra en la Figura MAEDH03, se procede a abrir la App y acceder a las funciones de la misma como lo ilustra la Figura MAEDH04.

Figura MAEDH04. Acceso a la App Draw-Hánzǐ

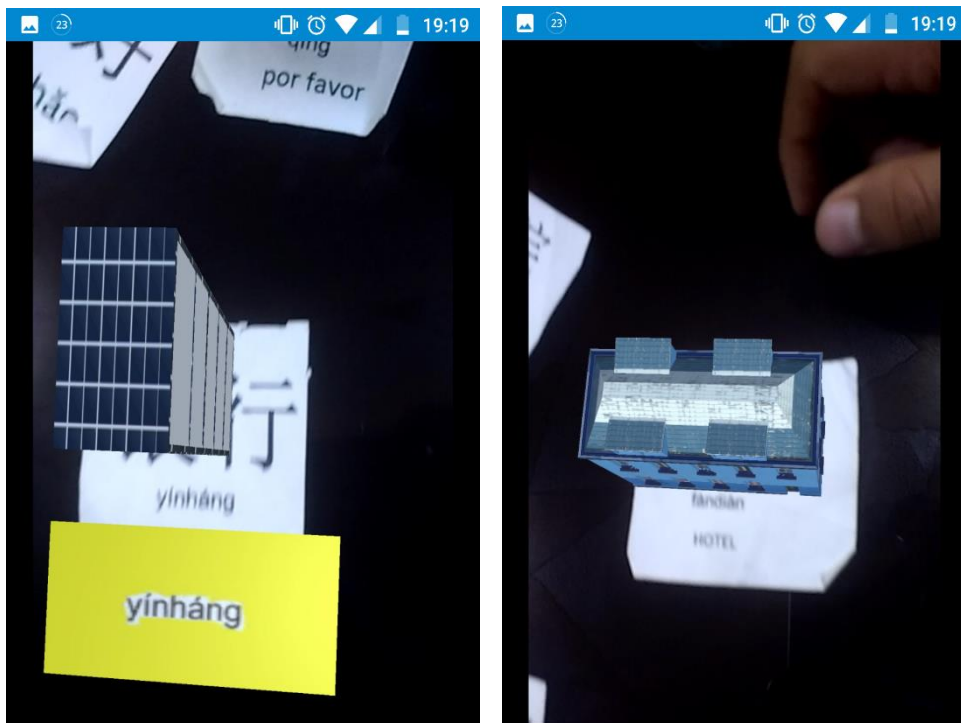


Fuente: Elaboración Propia, 2017.

4.1.4.1 Funcionalidad del App Draw-Hánzǐ

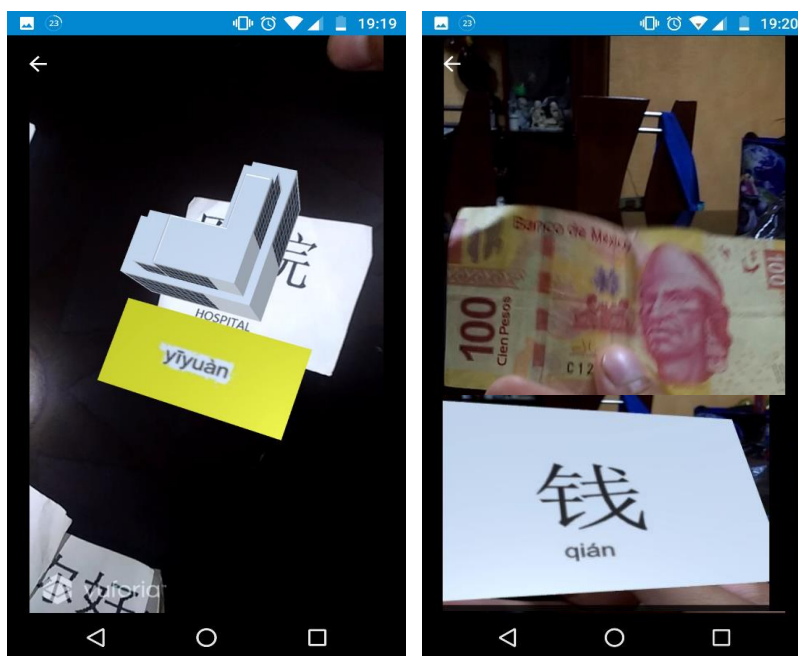
En las Figuras MAEDH05, MAEDH06 y MAEDH07, ilustran el funcionamiento de la App Draw-Hánzá con el uso de marcadores (Targets) y proyectando con el uso de la tecnología de realidad aumentada (AR, por sus siglas en el idioma inglés), imágenes 3D, Textos 2D y sonido. La Figura MAEDH08 muestra la hoja de marcadores utilizados para dicho funcionamiento.

Figura MAEDH05. Ejemplo #01 de la función de proyección de texto, imagen 3D y Sonido.



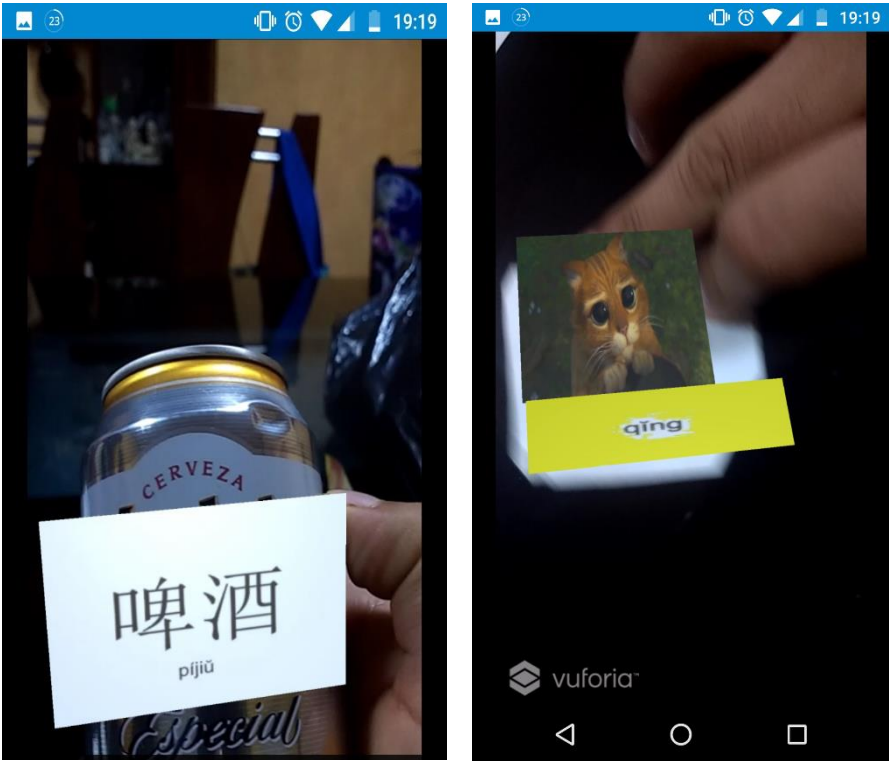
Fuente: Elaboración propia, 2017.

Figura MAEDH06. Ejemplo #02 de la función de proyección de texto, imagen 3D y sonido.



Fuente: Elaboración propia, 2017.

Figura MAEDH07. Ejemplo #03 de la función de proyección de texto, imagen 3D y sonido.



Fuente: Elaboración proopia, 2017.

Figura MAEDH08. Hoja de marcadores (Tarjets) utilizados y que la App reconoce perfectamente.

医生 谢谢 医院

yīshēng

DOCTOR

xièxie

GRACIAS

yīyuàn

HOSPITAL

厕所 请 你好 银行

cèsuǒ

BAÑO

qǐng

por favor

Nǐ hǎo

HOLA

yínháng

BANCO

帮助 饭店 食品

bāngzhù

AYUDA

fàndiàn

HOTEL

shípǐn

COMIDA



UAEM

Universidad Autónoma
del Estado de México

CENTRO UNIVERSITARIO UAEM TEXCOCO

“Aplicación de la Metodología Mobile-D™,
para la creación del prototipo de App
Caso de estudio: Draw-HánzÍ, sustentada en AR.”

ANEXO V

Cronograma General de desarrollo del Proyecto.



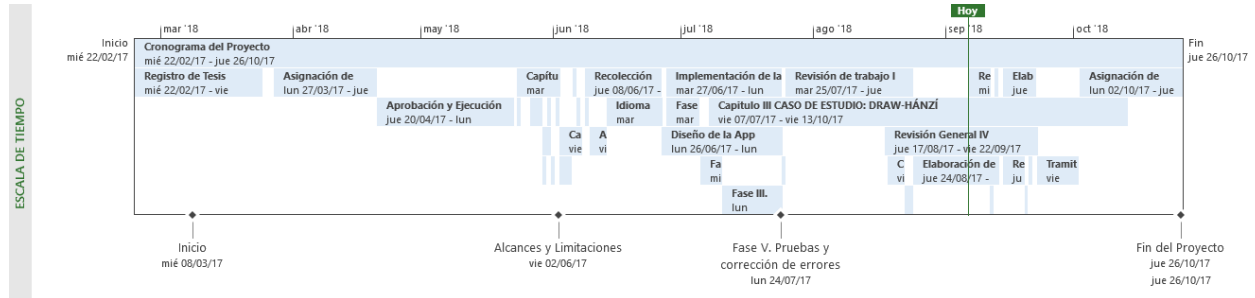
HUMANISMO
QUE TRANSFORMA

TEXCOCO, EDO. MÉX., OCTUBRE 2017

5. CRONOGRAMA GENERAL DE DESARROLLO DEL PROYECTO.

5.1. Línea del Tiempo del cronograma general.

Figura A01. Línea del tiempo general



Fuente: Elaboración propia, 2017.

3.2. Gráfica de Gantt del cronograma general.

Figura B01. Gráfica de Gantt parte I.

	M de ta	Nombre de tarea	Duración	Comienzo	Fin
1	→	Inicio	0 días	mié 08/03/17	mié 08/03/17
2	→	Registro de Tesis	18 días	mié 22/02/17	vie 24/03/17
3	→	Asignación de Asesores	15 días	lun 27/03/17	jue 20/04/17
4	→	Aprobación y Ejecución de Trabajo Escrito	16 días	jue 20/04/17	lun 22/05/17
5	→	Capítulo I FORMULACIÓN DEL PROYECTO	7 días	mar 23/05/17	vie 02/06/17
6	→	Introducción	1 día	mar 23/05/17	mié 24/05/17
7	→	Planteamiento del Problema	1 día	vie 26/05/17	lun 29/05/17
8	→	Objetivos	2 días	lun 29/05/17	mié 31/05/17
9	→	Objetivo General	1 día	lun 29/05/17	mar 30/05/17
10	→	Objetivo Específico	1 día	mar 30/05/17	mié 31/05/17
11	→	Hipótesis	1 día	mié 31/05/17	jue 01/06/17
12	→	Justificación	1 día	jue 01/06/17	vie 02/06/17
13	→	Alcances y Limitaciones	0 días	vie 02/06/17	vie 02/06/17
14	→	Capítulo II ESTADO DEL ARTE	3.2 días	vie 02/06/17	mié 07/06/17
15	→	Metodologías de Desarrollo de Software	1 día	vie 02/06/17	lun 05/06/17
16	→	Ingeniería de Software Móvil	1 día	lun 05/06/17	mar 06/06/17
17	→	Realidad Aumentada	1 día	mar 06/06/17	mié 07/06/17
18	→	Capítulo III CASO DE ESTUDIO: DRAW-HÁNZÍ	56 días	vie 07/07/17	vie 13/10/17
19	→	Recolección de Información	10 días	jue 08/06/17	lun 26/06/17
20	→	Apps Educativas	1 día	jue 08/06/17	vie 09/06/17
21	→	Aprendizaje Inmersivo	2 días	vie 09/06/17	mar 13/06/17
22	→	Idioma Chino	8 días	mar 13/06/17	lun 26/06/17
23	→	Diseño de la App Draw- Hanzí	16 días	lun 26/06/17	lun 24/07/17
24	→	Implementación de la Metodología Mobile-D	15 días	mar 27/06/17	lun 24/07/17
25	→	Fase I. Exploración	5 días	mar 27/06/17	mié 05/07/17

Fuente: Elaboración propia, 2017.

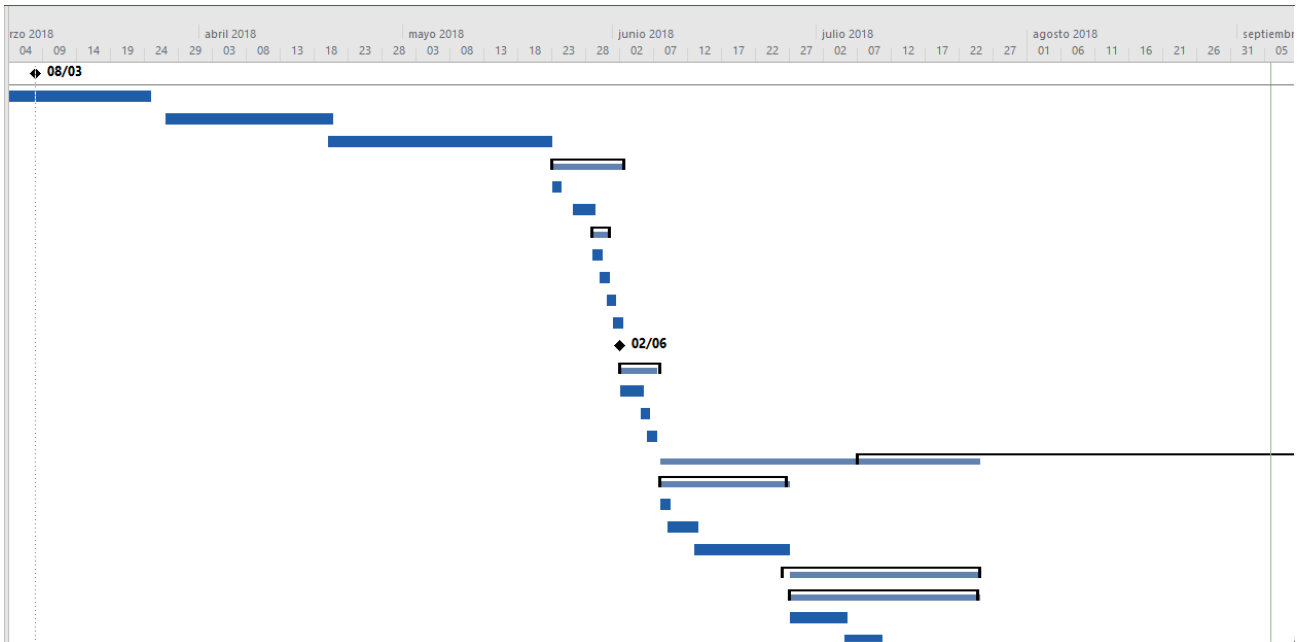
Figura B02. Gráfica de Gantt parte II.

	M de ta	Nombre de tarea	Duración	Comienzo	Fin
26	→	Fase II. Inicialización	3 días	mié 05/07/17	lun 10/07/17
27	→	Fase III. Producción	8 días	lun 10/07/17	lun 24/07/17
28	→	Fase IV. Estabilización	0 días	lun 24/07/17	lun 24/07/17
29	→	Fase V. Pruebas y corrección de errores	0 días	lun 24/07/17	lun 24/07/17
30	→	Resultados y Conclusión	1 día	lun 24/07/17	mar 25/07/17
31	★	Revisión de trabajo I	18 días	mar 25/07/17	jue 24/08/17
32	→	Cambios y Modificaciones	2 días	vie 18/08/17	mar 22/08/17
33	→	Corrección de Errores	2 días	mar 22/08/17	jue 24/08/17
34	★	Elaboración de Manual para la Creación de Apps	12 días	jue 24/08/17	mié 13/09/17
35	→	Revisión de Manual II	3 días	mié 06/09/17	lun 11/09/17
36	→	Cambios y Modificaciones	1 día	lun 11/09/17	mar 12/09/17
37	→	Corrección de Errores	1 día	mar 12/09/17	mié 13/09/17
38	★	Elaboración de Presentación para Sustentación de Proyecto de Tesis	5 días	jue 14/09/17	vie 22/09/17
39	→	Revisión de Presentación III	3 días	jue 14/09/17	mar 19/09/17
40	→	Cambios y Modificaciones	1 día	mar 19/09/17	mié 20/09/17
41	→	Corrección de Errores	1 día	mié 20/09/17	jue 21/09/17
42	→	Revisión General IV	21 días	jue 17/08/17	vie 22/09/17
43	→	Tramitología Correspondiente	5 días	vie 22/09/17	lun 02/10/17
44	→	Asignación de Fecha de Sustentación	15 días	lun 02/10/17	jue 26/10/17
45	★	Presentación y Sustentación	0 días	jue 26/10/17	jue 26/10/17
46	→	Fin del Proyecto	0 días	jue 26/10/17	jue 26/10/17

Fuente: Elaboración propia, 2017.

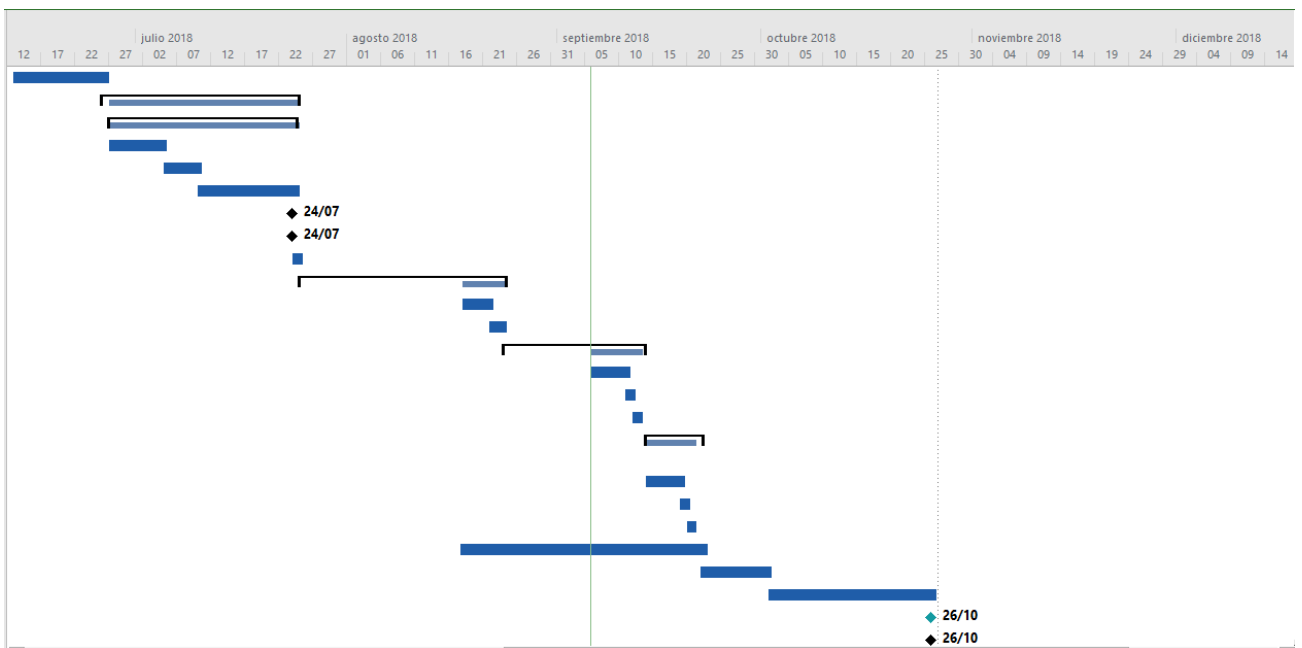
3.3. Gráfica de actividades por semana del cronograma general.

Figura C01. Gráfica de actividades parte I.



Fuente: Elaboración propia, 2017.

Figura C02. Gráfica de actividades parte II.



Fuente: Elaboración propia, 2017.



UAEM

Universidad Autónoma
del Estado de México

CENTRO UNIVERSITARIO UAEM TEXCOCO

“Aplicación de la Metodología Mobile-D™,
para la creación del prototipo de App
Caso de estudio: Draw-HánzÍ, sustentada en AR.”

APÉNDICE

I

**Metodología ágil de desarrollo de software para
aplicaciones móviles Mobile-D™.**



HUMANISMO
QUE TRANSFORMA

TEXCOCO, EDO. MÉX., OCTUBRE 2017

1. METODOLOGÍA MOBILE-D™.

1.1. Antecedentes.

La metodología Mobile-D™, fue creada en un proyecto de “una bicicleta de montaña” en Finlandia por el periodo del 2004-2005 (sin embargo, sigue vigente en algunos proyectos a nivel mundial); misma que se cataloga dentro de las metodologías de desarrollo ágil de software, para aplicaciones móviles (*Apps*, por sus siglas en ingles). Además de tener una vital importancia en el desarrollo de software móvil. Es bastante conveniente y recomendable, sí se observa desde varios contextos, por ejemplo: seguridad, financiera, logística y aplicaciones de productos de simulación.

Ha sido desarrollado en cooperación con tres compañías de software para móviles. Mobile-D™, ha sido evaluado con éxito en contra de la certificación *CMMI (Capability Maturity Model Integration)* nivel 2 y está continuamente más elevados, así mismo es validado en cursos *OOPSLA (Object - Oriented Programming, Systems, Languages and Applications)*, desde el 2004 y en diversos casos de estudio, así como proyectos en el *VTT Electronics (Centro de Investigación Técnica de Finlandia Ltd)*.

Es creada, como los mismos creadores lo definen “Un enfoque de desarrollo ágil”, ante el complicado y difícil, desarrollo de aplicaciones móviles, con la firme intención de superar los desafíos que presentan las específicas demandas del desarrollo móvil y las limitaciones técnicas. Es sustentado principalmente en la experiencia adquirida a partir de cuatro estudios de caso en *VTT Electronics* y del amplio conocimiento de sus creadores.

Es presentado en el *OOPSLA '04* en su edición número 19, el 02 de Noviembre del 2004 en Vancouver, BC. por sus creadores:

- *Pekka Abrahamsson*, Centro de Investigación Técnica VTT de Finlandia.
- *Antti Hanhieva*, Centro de Investigación Técnica VTT de Finlandia.
- *Hanna Hulkko*, Centro de Investigación Técnica VTT de Finlandia.
- *Tuomas Ihme*, Centro de Investigación Técnica VTT de Finlandia.
- *Juho Jaalinoja*, Centro de Investigación Técnica VTT de Finlandia.
- *Mikko Korkala*, Universidad de Oulu.
- *Juha Koskela*, Centro de Investigación Técnica VTT de Finlandia.
- *Pekka Kyllonen*, Centro de Investigación Técnica VTT de Finlandia.

- *Outi Salo*, Centro de Investigación Técnica VTT de Finlandia.

Quienes contribuyeron en la redacción y creación de cada una de sus fases, etapas, tareas, métricas y formato de documentación, respectivamente, misma que se detalla en lo general más adelante en este documento. Todo esto con base y a través de su experiencia en el desarrollo de software.

Actualmente los *OOPSLA* forman parte ya de los *SPLASH (Systems, Programming, Languages, and Applications: Software for Humanity)*, eventos a nivel mundial, que incluyen todos los aspectos de los lenguajes de programación e ingeniería de software, en un sentido bastante amplio. Los trabajos que aborden cualquier aspecto de desarrollo de software, son bienvenidos en este evento, mismo que abarcan los temas desde: Requisitos, modelado, creación de prototipos, diseño, implementación, generación, análisis, verificación, evaluación, mantenimiento, reutilización, sustitución y retirada de sistemas de software. Los trabajos pueden abordar estos temas y muchos más, en una variedad de maneras, incluyendo nuevas herramientas (cómo idiomas, análisis de programas, y los sistemas en tiempo de ejecución), las nuevas técnicas (cómo las metodologías, los procesos de diseño, métodos de organización de código y técnicas de gestión), y nuevas evaluaciones (cómo formalismos y pruebas, análisis de corpus, estudios de usuarios y encuestas).

Dentro de los múltiples atractivos que ofrece el evento a los profesionales, investigadores y estudiantes de diversas disciplinas que tienen en común la tecnología, incubando y practicando el estado de la técnica tecnológica están los siguientes:

- Documentos Técnicos: Revisados rigurosamente, documentación que contribuye a los fundamentos de la tecnología de objetos.
- OnWard: Una pista técnica de comunicaciones que presentan nuevas ideas y nuevos paradigmas de la computación.
- Paneles: Pequeños grupos de expertos que debaten cuestiones de la tecnología de objetos, con la participación de la audiencia.
- Informes Prácticos: Informes desde el campo que describe la experiencia del mundo real con la tecnología de objetos.
- Tutoriales: Cursos a cargo de expertos de renombre internacional en la amplia gama de temas relacionados con los objetos. Las clases particulares se asignan en tres niveles: iniciación, intermedio o avanzado.

- Talleres: Entornos intensivos, de colaboración donde los participantes salen a la superficie, discuten y resuelven problemas desafiantes que enfrentan en el campo laboral.
- Posters: Presentaciones de trabajo informales de sondeo, en un entorno visual e interactivo.
- Concurso de Investigación de Estudiantes: el concurso de investigación de estudiantes de *ACM SIGPLAN*.
- Manifestaciones: Mostrar y contar las sesiones para los sistemas del estado de la técnica, lenguajes, entornos y aplicaciones.
- *DesignFest*: Pequeños grupos de diseñadores y desarrolladores trabajan en problemas de análisis y diseño.
- Simposio de Educadores: Un foro de un día en que los educadores universitarios y la formación de expertos del sector comparten experiencias, problemas y técnicas.
- Doctorado Simposio: Una sesión de un día en que un panel de expertos proporciona orientación y retroalimentación a Ph. D. y estudiantes en sus temas de tesis.
- Exposiciones: Muestra y demostración de los últimos productos de las compañías de software, consultores, especialistas en formación, los vendedores de aparatos electrónicos, personales y los editores.
- Estudiantes Voluntarios: Al donar un par de horas de su tiempo, los estudiantes pueden asociar con los máximos de la tecnología orientada a objetos, la investigación y el desarrollo de software.

Entre muchas otras, que se agregan con las posteriores ediciones.

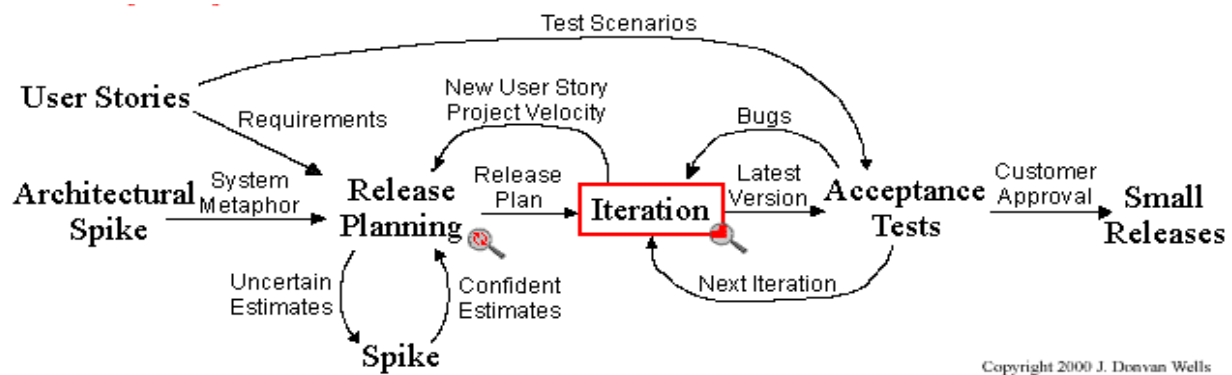
1.2. Objetivos de la metodología Mobile-D™.

El objetivo de esta metodología es conseguir ciclos de desarrollo muy rápidos en equipos muy pequeños o con mira a grupos de pocas personas (con un máximo de 10 personas), los integrantes del equipo de desarrollo deben poseer una habilidad y capacidad similar, entre ellos (Centro de Investigación Técnico VTT de Finlandia, s.f.).

1.3.Bases.

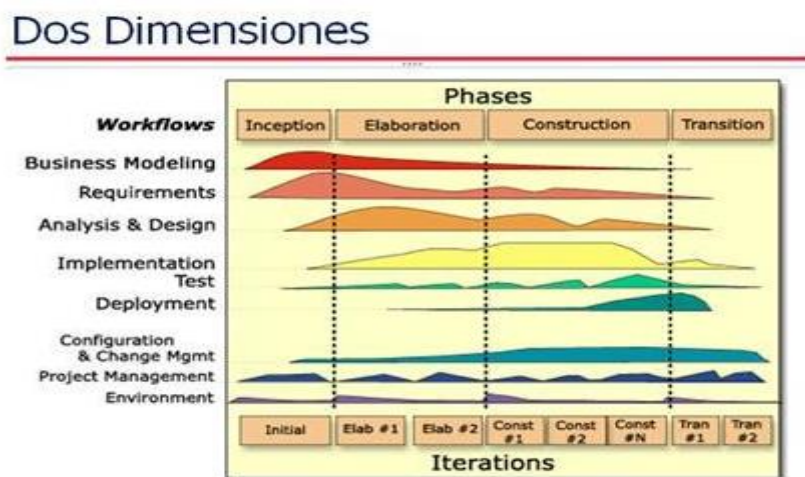
Mobile-D™ se basa en soluciones metodológicas conocidas y consolidadas como lo son: programación extrema (XP, por sus siglas en el idioma inglés), para las prácticas y el desarrollo, ilustrada en la Figura AP01 ; metodologías cristal clear (CC, por sus siglas en el idioma inglés) para la escalabilidad y los métodos, como lo muestra la Figura AP02 y el proceso unificado racional (RUP, por sus siglas en el idioma inglés), ilustrada en la Figura AP03, para la cobertura y como base del diseño del ciclo de vida.

Figura AP01. Proyecto con Extreme Programming



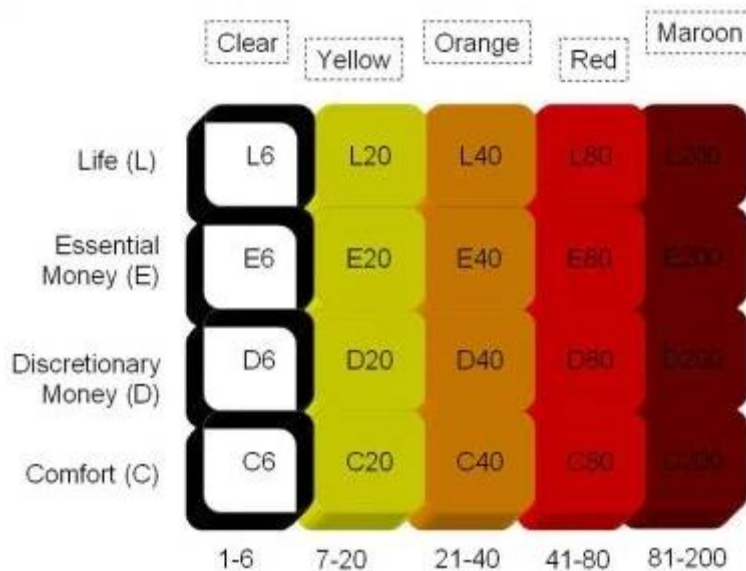
Fuente: Don Wells, XProgramming.com, 2000.

Figura AP02. Fases del ciclo de vida RUP.



Fuente: Procesos de Software, 2010.

Figura AP03. Ciclo de vida de la metodología de CC.



Fuente: Fuentes López, G., Temoltzin Moreno, F., Rodriguez Briones, L., Bonilla Corona, J. y López Hernández, R., 2013.

Se basa en los nueve elementos principales siguientes:

- FASE Y RITMO

Los proyectos se llevan a cabo en iteraciones, cada una de las cuales comienza con una reunión y una planificación del día.

- LÍNEA DE ARQUITECTURA

Enfoque de la línea de arquitectura se utiliza junto con los patrones arquitectónicos y el modelado ágil.

- DESARROLLO IMPULSADO POR PRUEBAS MÓVILES

El método de prueba primero se utiliza junto con los casos de prueba automatizados.

- INTEGRACIÓN CONTINUA

Las prácticas de SCM eficaces se aplican a través de múltiples medios.

- PROGRAMACIÓN EN PAREJAS

Codificación, pruebas y refactorización se llevan a cabo en parejas.

- MÉTRICAS

Pocos parámetros de medición esenciales se recogen rigurosamente y se utilizan para la retroalimentación y fines de mejora de procesos.

- MEJORA DE PROCESOS DE SOFTWARE ÁGIL

Los talleres posteriores a la iteración se utilizan para mejorar continuamente el proceso de desarrollo.

- CLIENTE FUERA DEL SITIO

Al cliente participa en los días de planificación y la liberación.

- ENFOQUE CENTRADO EN EL USUARIO

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

1.4.Glosario Mobile-D™

Nombre del Documento: Mobile-D™ Glosario.

Creado por: Outi Salo & Juha Koskela.

Fecha: 29-11-2004.

Este documento enumera el vocabulario y la variedad de insumos, productos y funciones. Utilizados en la biblioteca de procesos Mobile-D. Además, la descripción de cada elemento es incluido en este glosario.

Las principales Entradas y salidas:

1. LOS DOCUMENTOS DE PRUEBA DE ACEPTACIÓN: es un documento que contiene las descripciones de las pruebas de aceptación.
2. PRUEBA DE ACEPTACIÓN: Es una prueba formal llevada a cabo para determinar si el sistema satisface o no, los criterios de aceptación y permite al cliente determinar si acepta el sistema propiamente.
3. LISTA DE PUNTOS DE ACCIÓN: se crea en el taller después de la iteración. La lista incluye, por ejemplo, la información sobre temas de mejoras, resultados, responsabilidades y plan de seguimiento.
4. DESCRIPCIÓN DE LA LÍNEA DE ARQUITECTURA: es creado por el grupo de soporte y/o apoyo durante la fase de exploración y proporciona un fondo arquitectónico útil, información y propuestas (no especificaciones) sobre el sistema a hacer. Es cuando el equipo desarrollador del proyecto crea un plan de línea de arquitectura, arquitectura y diseño de software, para la descripción del proyecto.
5. PLAN DE LÍNEA DE ARQUITECTURA: se creó para el proyecto por el equipo desarrollador del proyecto, dentro del alcance de los requerimientos del cliente. El plan puede ser incluido en el plan del proyecto.
6. DESCRIPCIÓN DEL PROCESO DE BASE: es el proceso que se adapta y basa en la biblioteca de procesos Mobile-D y describe el proceso organizativo del proyecto inicial. Esta descripción se adapta iterativamente y se actualiza a lo largo del proyecto con la técnica del taller post-iteración.
7. CONTRATO: es el acuerdo entre el cliente y la organización de desarrollo de software. Es el documento donde se acuerda el cronograma del proyecto, el alcance y las finanzas, se acuerdan. En Mobile-D, la idea básica es establecer el horario fijo para el proyecto, donde el alcance son los que están acordando (es decir, los requerimientos de bajo nivel no se implementaran, sí estos superan el cronograma y será necesario otro contrato para su aplicación).

8. **INFORME DE ESTADO DIARIO:** es un documento que se puede utilizar para informar al cliente. Describe las tareas, los posibles problemas y soluciones, así como una gran variedad de cuestiones diversas.
9. **LISTA DE DEFECTOS:** es un documento que incluye descripciones de los defectos encontrados.
10. **OBSERVACIONES DE LOS DESARROLLADORES:** se recomienda a los desarrolladores anotar, por ejemplo, discusiones con el cliente. Las notas pueden ser valiosas cuando se desarrolle el sistema.
11. **IMPLEMENTACIÓN DE FUNCIONALIDAD:** este término se refiere a una pieza funcional del producto (por ejemplo, una cierta parte del código). Este término también incluye código fuente, código de programa, código de prueba, línea de base y sistema /producto.
12. **INFORMACIÓN DEL RADIADOR:** contiene historias del usuario y tareas de una iteración. El visionario puede ver los estados de las historias, tareas e iteración.
13. **DOCUMENTO DE REQUISITOS INICIALES:** es la propuesta del producto definiendo los requisitos centrales del producto, tanto funcionales como no funcionales, se definen con suficiente detalle para posibilitar la planificación inicial del proyecto. El grupo del cliente es la parte clave, en esta parte de los requisitos.
14. **ITERACIÓN:** es un ciclo de diseño / desarrollo. Por ejemplo, producir iteración consiste en días de planificación, trabajo y liberación.
15. **PLAN DE MEDIACIÓN:** se describen los parámetros que deben recogerse en el proyecto, los medios para recogerlos, su validación, así como el plan para su uso (por ejemplo, monitoreo de proyectos, aprendizaje, etc.).

16. MÉTRICA DE DATOS: se utilizan en la fase de producción como información, que es recolectada de iteraciones anteriores.
17. BIBLIOTECA DE PROCESOS MOBILE-D™: es el conjunto de patrones Mobile-D™, que consisten en patrones de fase, etapas, tareas y nivel de prácticas.
18. BIBLIOTECA DE PROCESOS DE LA ORGANIZACIÓN: es el proceso (biblioteca y políticas) que se encuentra actualmente en uso en la organización que está adoptando Mobile-D™ y/o sus partes.
19. PILA DE PRODUCTO: es una herramienta para la gestión de los requisitos del producto durante las actividades de desarrollo de software. El backlog se puede implementar en (por ejemplo, una plantilla de documento o herramienta de base de datos), dependiendo del contexto organizacional. El backlog se utiliza para que el cliente administre los requisitos (funcionales y no funcionales), así como las historias y las tareas que se definen sobre la base de ellos.
1. Historia: Es básicamente una cosa que el cliente quiere que el sistema haga, también conocido como “Historias de Usuarios”.
 2. Tareas: el equipo de desarrollo divide las historias en tareas y actividades, que son descripciones técnicas del trabajo.
20. PROPUESTA DE PRODUCTO: es un documento que contiene la visión inicial del producto de software. La propuesta es la entrada para la fase de exploración del proyecto y puede, en esta etapa, estar en una descripción del nivel bastante alto, que da una idea general del producto.
21. PLAN DEL PROYECTO: es un documento que describe el propósito del proyecto, define los recursos y la escala de tiempo para el proyecto. El plan del proyecto es actualizado iterativamente durante el proyecto y se crea en la fase de exploración.

22. **LIBERACIÓN:** en este contexto la liberación es la iteración final, en la etapa final del producto.
23. **LISTA DE APROBACIÓN DE SALIDA:** es un documento que contiene las cosas que deben comprobarse en relación con cada liberación del producto.
24. **RITMO:** cada proyecto tiene cierto ritmo. Definir el ritmo significa una etapa donde las longitudes y el número (estimado) de iteraciones (es decir, lanzamientos de productos).
25. **DESCRIPCIÓN DE LA ARQUITECTURA Y DISEÑO DE SOFTWARE:** se crea un documento, por parte del equipo desarrollador del proyecto, que transmite información arquitectónica y de diseño sobre el software desarrollado de manera completa, consistente y formal a manera que otras personas que el equipo desarrollador del proyecto, puedan entender lo que fue construido, ¿Por qué? fue construido de esa manera, y ¿Cómo? mantenerlo.
26. **PICOS:** se crean para averiguar las respuestas a las difíciles características técnicas o problemas estéticos. El código de “Pico” suele ser desechado después de las “Soluciones Potenciales”.
27. **NORMAS Y POLÍTICAS:** estas podrían ser relevantes durante el proceso de desarrollo.
28. **INFORME DE LA PRUEBA DEL SISTEMA:** es una breve descripción del proceso de prueba y lista detallada de defectos encontrados.
29. **REGISTRO DE LAS PRUEBAS:** es un documento que contiene todos los casos de prueba; que se ejecutaron en las pruebas del sistema y los resultados de estas mismas.

30. MATERIAL DE FORMACIÓN: es proporcionada por el personal de formación (Grupo de soporte y/o apoyo), que planificará y conducirá la capacitación para el equipo desarrollador del proyecto, desde el comienzo del proyecto.
31. PLAN DE FORMACIÓN: se definen las necesidades de formación del equipo de desarrollo del proyecto.
32. ILUSTRACIONES DE INTERFAZ DE USUARIO: describen puntos de vista de los desarrolladores de la interfaz de usuario, a las que el cliente puede y debe dar retroalimentación constante, para la realización de las ilustraciones de interfaz de usuario.
33. UNIDAD DE PRUEBA: mide una unidad de software. Las pruebas unitarias se crean a más tardar, el código es refactorizado.
34. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO: contiene la interfaz de usuario del sistema. Incluye las descripciones y la funcionalidad definida por el usuario.

Los siguientes roles o personajes (grupo de interesados), se pueden identificar en la ejecución del proceso de Mobile-D™:

1. GRUPO DE CLIENTES: se compone de personas que son muy conscientes de los requisitos para el producto de software que se está desarrollando. También estos miembros, del grupo de clientes deben tener autoridad adecuada y/o suficiente, para cambiar los requisitos del producto durante el desarrollo del proyecto, deben tener sus funciones claramente definidas (por ejemplo, el papel del cliente) y las tareas para participar en ciertas actividades de desarrollo de software (por ejemplo, la planificación, la priorización de las tareas y las pruebas de aceptación) y el grupo de participantes suficientes para proporcionar una amplia experiencia en diferentes aspectos de los requisitos del producto (funcional y no funcional) al equipo de desarrollo del proyecto.

2. **EXPLORACIÓN DEL EQUIPO:** es un grupo que se necesita para llevar a cabo la fase de exploración del proyecto. En esta etapa el equipo desarrollador del proyecto está ausente.
3. **EQUIPO DESARROLLADOR DEL PROYECTO:** está formado por los desarrolladores de software y la administración del proyecto, las actividades de desarrollo del software real.
 1. Equipo del proyecto / Arquitecto
 2. Equipo del proyecto / Tester
 3. Equipo del proyecto / Colección de Métricas
4. **GRUPO DE DIRECCIÓN:** se compone de diferentes partes interesadas de un proyecto que tienen la autoridad para tomar decisiones y monitorear el progreso del proyecto, por ejemplo, el cliente y los miembros del equipo de exploración, pueden ser parte del grupo directivo.
5. **GRUPO DE SOPORTE Y/O APOYO:** se compone de varias funciones diferentes que son necesarios durante todo el desarrollo del proyecto. Algunos sub-roles del personal del grupo de soporte y/o apoyo, pueden, por ejemplo, ser el arquitecto, el especialista en procesos, formadores, así como varios expertos de diferentes áreas de dominio. Las tareas de los diferentes miembros del grupo de soporte y/o apoyo, pueden incluir actividades de aseguramiento de la calidad, seguimiento de proyectos, análisis de métricas y coaching durante el proceso de desarrollo del proyecto.
6. **GRUPO DE PRUEBAS DEL SISTEMA:** es un grupo de personas asignados para realizar pruebas (System Testing), ya que no es factible para los desarrolladores probar su propio software.
7. **MODERADOR:** tiene un papel importante, el de dirigir el taller de post-iteración. Las tareas del moderador incluyen la agrupación del calendario de mantenimiento y la discusión en pista. La persona adecuada para el papel de moderador es alguien fuera del equipo de desarrollo del proyecto y que debe centrarse en la generación de

experiencias y mejoras. El moderador también debe ser un enlace entre el nivel del proyecto y comunicación a nivel organizacional.

8. AUDITOR EXTERNO: es una persona que pasa a través de la auditoría de liberación, la lista de verificación y anota las deficiencias detectadas, Se asegura de que el / los resultados de la auditoría de publicación son válidos y aceptables. El auditor debe tener un buen conocimiento del proyecto.

9. SECRETARIO: tiene la tarea de crear y mantener actualizada la lista de puntos de acción. Además de las listas de punto de acción anterior, deberá hacer el seguimiento y la actualización pertinente.

1.5.Fases de la metodología Mobile-D™.

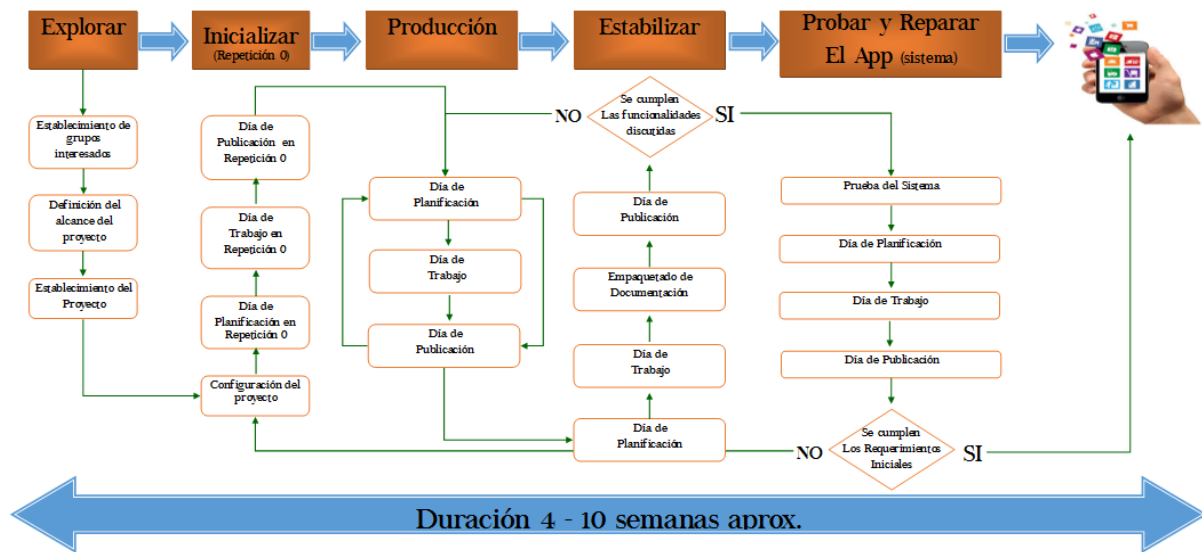
La metodología Mobile-D™, se compone de distintas fases: exploración, inicialización, producción, estabilización y pruebas. Cada una tiene un día de planificación y otro de entrega y/o liberación (con excepción de la fase de exploración), ilustrada en la Figura AP04, a continuación:

Figura AP04. Fases de la metodología Mobile-D™



Fuente: Centro de Investigación Técnico VTT de Finlandia, 2004-2005.

Figura 3. Diagrama de procesos de la metodología ágil de desarrollo de software para aplicaciones móviles Mobile-D™.



Fuente: Elaboración propia, 2017.

1.5.1. Fase de EXPLORACIÓN.

Se centra la atención en la planificación y los conceptos básicos del proyecto, ilustrado en la Figura AP05. Aquí es donde se hace una definición del alcance del proyecto y su establecimiento con las funcionalidades, de a dónde se quiere llegar.

Figura AP05. Ejemplo de fase de exploración.



Fuente: Op. Cit., 2004-2005.

Nombre: Exploración.

Autor: Outi Salo & Hanna Hulkko.

Adaptación y/o Modificación: Juha Koskela.

Fecha: 01-01-2004

Tipo de patrón: Patrón de fase

El propósito de esta fase de exploración es la planificación y establecimiento del proyecto inicial “*A well planned is half done* (Sí hay una buena planificación, ya está la mitad del trabajo hecho)”, es un dicho que recuerda el contexto del desarrollo de software. La fase de exploración puede ser oportuna, sin estar unida, a las últimas fases de Mobile-DTM y también se superpone con la fase de inicialización. La fase de exploración es muy importante para establecer las bases de una implementación de software/producto bien controlada, con respecto, por ejemplo, a cuestiones relacionadas con la arquitectura del producto, el proceso de desarrollo de software y la selección del medio ambiente. Se necesitan diferentes puntos de vista dentro del equipo de desarrollo y de las partes interesadas en el producto, para ofrecer una mejor experiencia en la fase de exploración.

Los objetivos de esta fase de exploración son:

- I. Establecer y definir los grupos de interés y/o personajes interesados que intervienen en la planificación y el seguimiento del proyecto de desarrollo.
- II. Definir y acordar las metas, los alcances y límites del proyecto de desarrollo.
- III. Planificar el proyecto en relación con el medio ambiente, el personal y los problemas del proceso.

Los criterios de entrada en la fase de exploración son:

1. La propuesta del producto existente.
2. El grupo de exploración para el inicio del proyecto (estos ya han sido definidos previamente).
3. Se genera un compromiso con la administración para la realización del proyecto.

Las entradas de la fase de exploración son:

1. Propuesta del producto.
2. Biblioteca de procesos Mobile-D™.
3. Biblioteca de procesos organizativos.
4. Contrato.
5. Documento de requisitos iniciales.
6. Plan de proyecto.
7. Normas y restricciones en el que caso de que existan (políticas de la empresa).

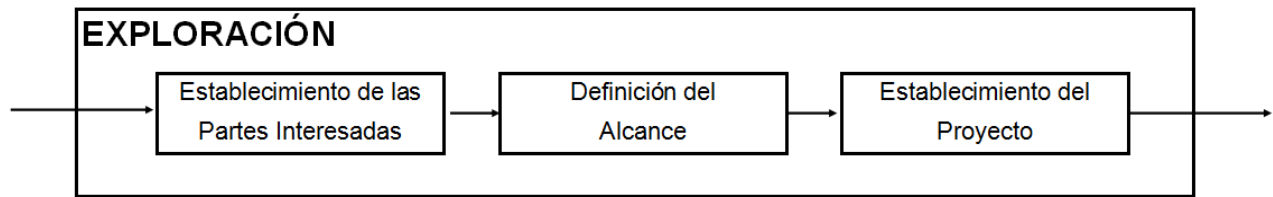
Los criterios de salida de la fase de exploración:

1. El grupo del cliente ha sido identificado.
2. Los grupos de interés y/o personajes interesados en el desarrollo del proyecto ya se han definido y establecido.
3. Los requisitos iniciales y el calendario se han definido para el producto.
4. El personal y el medio ambiente, necesario durante el desarrollo de productos (Se han seleccionado y asignado, por ejemplo, ayuda del equipo / capacitación del personal / entrenadores, grupo de dirección / gestión, atención al cliente).
5. La descripción del proceso ha sido adaptada y documentada para el incipiente proyecto.
6. Plan de entrenamiento.

Las salidas de la fase de exploración, ilustrada en la Figura AP06:

- Documento de requisitos iniciales donde se han definido los requerimientos iniciales del desarrollo del producto.
- Plan de proyecto incluyendo línea de tiempo, el ritmo, las terminaciones, los recursos del proyecto, los actores, los roles y sus responsabilidades.
- Descripción base del proceso que incluye la línea de base, las actividades de seguimiento de calidad, documentación, puntos de integración del hardware a llegar a las salidas.
- Plan de medición.
- Plan de formación y/o entrenamiento.
- Descripción de la línea de la arquitectura.

Figura AP06. Etapas de la fase de exploración



Fuente: Op.Cit., 2004-2005.

Las etapas de la fase de exploración son:

I. Establecimiento de las partes interesadas

El propósito de esta etapa es identificar y establecer los grupos de interés que se necesitan en diversas tareas de la fase de exploración, así como en el apoyo de las actividades durante el desarrollo del proyecto de software - excluyendo el equipo de desarrollo en sí-. Es necesaria una gran variedad de conocimientos y la cooperación para planificar una aplicación controlada y efectiva del producto de software.

- Establecimiento de actores: el propósito de esta tarea es reconocer que el grupo de clientes tiene la experiencia, conocimiento del dominio y de los requisitos para el desarrollo del producto de software.

II. Definición del alcance

El propósito de esta etapa es definir las metas para el inicio del proyecto, con respecto al contenido, así como al cronograma del mismo.

- Planeación inicial del proyecto: el propósito de esta tarea es establecer el plan inicial, de los procesos de desarrollo de software con respecto a una línea de tiempo, el ritmo y las inversiones del proyecto.
- Conjunto de requisitos iniciales: el propósito de esta tarea es la de producir una definición general inicial del alcance del producto, el propósito y la funcionalidad.

Esto es necesario la etapa del establecimiento del proyecto (tamaño, cuestiones, técnicas, arquitecturas, por mencionar algunas).

III. Establecimiento del proyecto

El propósito de esta etapa es definir y asignar los recursos (tanto técnicos, así como humanos), necesarios para el proyecto de desarrollo inicial del software. El proceso de línea de base para el proyecto es una tarea importante de esta etapa y también el establecimiento del proyecto. La fase de establecimiento es para asegurarse de que el equipo desarrollador del proyecto puede iniciar el desarrollo de software sin demoras causadas, por ejemplo, por la falta de herramientas y el entrenamiento apropiado.

- Definición de la arquitectura: el propósito de esta tarea es definir suficientemente, los posibles problemas de arquitectura para que el proyecto se puede realizar con éxito.
- Proceso de establecimiento: esta se construye desarrollando una serie de pasos.
 - Establecer la línea de proceso base.
 - Planificar la documentación necesaria.
 - Planificar el seguimiento y la medición.
 - Identificar las necesidades de formación.

Los siguientes roles están relacionados con la fase de exploración:

1. Equipo desarrollador del proyecto.
2. Grupo de soporte y/o apoyo.
3. Grupo del cliente / cliente.
4. Grupo de dirección.
5. Equipo de exploración.

La fase de exploración se compone de los siguientes patrones:

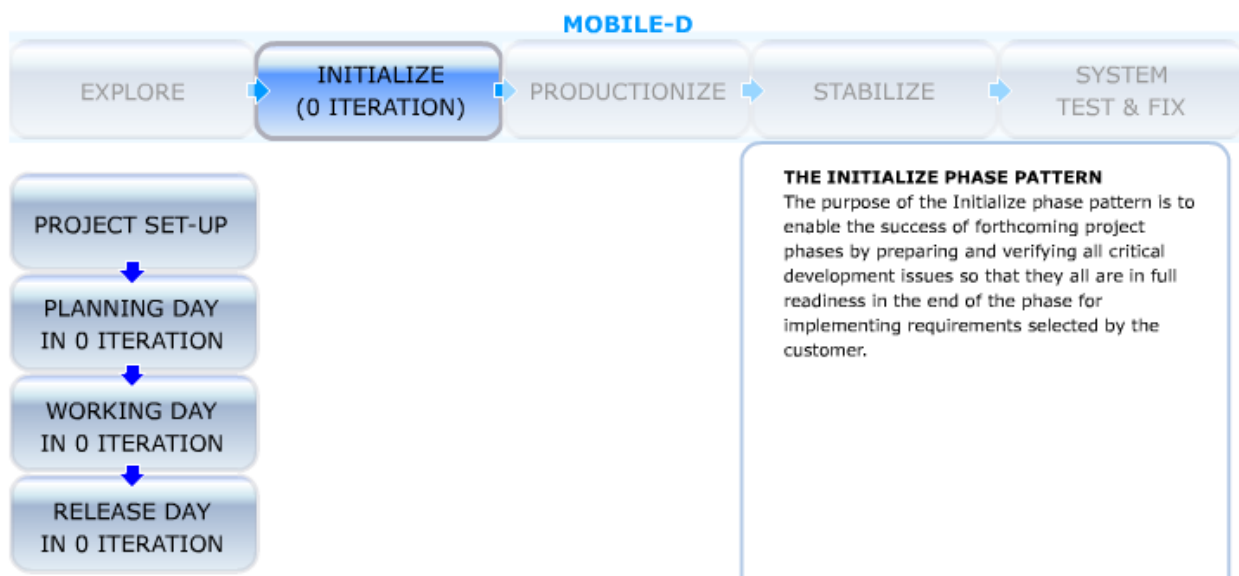
- **ESTABLECIMIENTO DE LOS INTERESADOS:** es la identificación y asignación de los grupos y partes interesadas pertinentes, son las primeras tareas de la fase de exploración.

- **DEFINICIÓN DEL ALCANCE:** objetivos y alcance del proyecto deberían ponerse de acuerdo sobre las etapas iniciales del proyecto, incluidos los requisitos, la planificación del proyecto y cuestiones arquitectónicas.
- **ESTABLECIMIENTO DEL PROYECTO:** la etapa del establecimiento del proyecto incluye la sección del medio ambiente para el proyecto, la asignación del personal y el proceso.
- **ESTABLECIMIENTO DE CLIENTES:** el patrón del establecimiento de tareas de atención al cliente, es una instancia del patrón del establecimiento de las partes interesadas. Debido al papel importante del cliente, ya sea dentro o fuera del establecimiento de la colaboración con el cliente está incluido en la descripción del patrón de Mobile-D™.

1.5.2. Fase de INICIALIZACIÓN.

En esta fase es donde se configura el proyecto, identificando y preparando todos los recursos necesarios, en esta fase se dedica un día a la planificación y el resto del trabajo y la publicación, como lo muestra la Figura AP007, a continuación:

Figura AP07. Ejemplo de fase de inicialización.



Fuente: Op. Cit., 2004-2005.

Autor: Tuomas Ihme

Fecha: 16-03-2005

Tipo de patrón: Patrón de fase

Clasificación de patrón: Esencial

El propósito de esta fase de inicialización es permitir el éxito de las siguientes fases del proyecto mediante la preparación y verificación de todas las cuestiones críticas fundamentales del desarrollo, todos están en plena preparación al final de la fase de implementación de los requisitos seleccionados por el cliente.

Los objetivos de esta fase de inicialización son:

1. Obtener una buena comprensión global del producto: por parte del equipo de desarrollo del proyecto, sobre los requisitos iniciales y la línea de la arquitectura.
2. Preparar los requisitos físicos, técnicos y humanos, así como la comunicación con el cliente, los planes del proyecto y todas las cuestiones fundamentales de desarrollo: de manera que los problemas están todos plenamente preparados para la aplicación de los requisitos, durante las próximas fases del proyecto.

Los criterios de entrada de la fase de inicialización son:

1. La fase presente de exploración se ha completado correctamente.

Las entradas de la fase de inicialización son:

1. Documento de requisitos iniciales.
2. Plan de proyecto y descripción del proceso base.
3. Base de descripción del proceso.
4. Plan de Medición.
5. Plan de formación.
6. Descripción de la línea de arquitectura.
7. Cartera de productos.

Los criterios de salida de la fase de inicialización son:

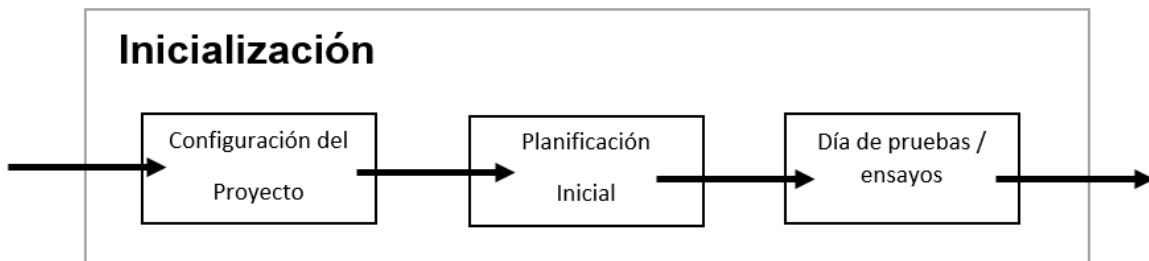
1. El plan del proyecto se ha actualizado y un plan de línea de arquitectura ha sido creado.
2. Se han realizado los controles previstos.

Las salidas de la fase de inicialización son:

1. Actualización del plan del proyecto y el plan de línea de arquitectura.
2. La primera versión de arquitectura y diseño de software, descripción del documento.
3. Funcionalidad implementada.
4. Primera versión de la cartera de productos.
5. El documento de requisitos iniciales actualizada.
6. El desarrollador observa y la interfaz de usuario-ilustraciones de cada requisito discutido.
7. Las pruebas de aceptación para cada requisito.

El patrón de la fase de inicialización, como lo muestra la Figura AP08, puede realizarse a través de las siguientes etapas:

Figura AP08. Ejemplo de la fase de inicialización.



Fuente: Op. Cit., 2004-2005.

Las etapas en individuales de la fase de inicialización son las siguientes:

I. Configuración del Proyecto.

El propósito de esta etapa es:

- 1) El establecimiento de los recursos físicos y técnicos: para la supervisión y control del proyecto, así como el monitoreo del medio ambiente del mismo.

- 2) Capacitación del equipo desarrollador: proporcionándoles el entrenamiento según sea necesario.
- 3) Establecer las formas específicas de comunicación del proyecto con el grupo de clientes: todas las tareas de la preparación del proyecto incluyen la participación del equipo.
 - Configuración del entorno: el procedimiento muy general y depende en gran medida del proyecto y del ambiente donde se va a desarrollar el proyecto, por lo tanto, en Mobile-D™ este procedimiento no es especificado como tal.
 - Formación: el procedimiento es muy general y depende en gran medida del proyecto y sus características, del ambiente, donde se va a desarrollar el proyecto, por lo tanto, en Mobile-D™ este procedimiento no es especificado como tal.
 - Planificación inicial: el propósito del modelo de planificación inicial es obtener una buena comprensión general del producto a desarrollar, para preparar y perfeccionar los planes para las próximas fases del proyecto y preparar planes de comprobación y resolución de todas las cuestiones fundamentales del desarrollo final de la fase actual.

II. Planificación inicial (Día de planificación en 0 iteraciones)

El propósito del patrón de planificación inicial es obtener una buena comprensión general del producto desarrollado, para preparar y refinar los planes para las próximas fases del proyecto y preparar planes para verificar y resolver todos los problemas críticos de desarrollo final, de la fase actual.

- Planteamiento de la arquitectura: el propósito de esta tarea es preparar los problemas críticos de la arquitectura de manera que estén a disposición cuando se haga la implementación de los requisitos del cliente.
- Análisis de los requisitos iniciales: el propósito de esta tarea es priorizar y analizar los requisitos. La búsqueda de un conjunto de requisitos obliga a crear componentes más importantes e interfaces del sistema. Un esqueleto

arquitectónico de trabajo no debe ser encontrado, después del final de la primera iteración.

III. Día de pruebas / ensayos

El propósito de esta etapa es probar y establecer aún más el desarrollo técnico, ambiente y asegurarse de que todo esté listo para implementar el desarrollo de productos de software. Además, el propósito es implementar algunas funcionalidades (por ejemplo, comunicación cliente-servidor) o resolver algún problema crítico de desarrollo, sin producir ningún código de trabajo. Otras investigaciones tecnológicas posibles en esta etapa. Sí el desarrollo decide implementar alguna funcionalidad en este punto, no es necesario que sea la funcionalidad de su importancia en relación, por ejemplo, la estructura arquitectónica del producto. Los días de prueba forman la pre-fase para los días de desarrollo real.

Los siguientes roles están relacionados con la fase de inicialización:

1. Equipo desarrollador del proyecto.
2. Equipo desarrollador del proyecto / gerente del proyecto.
3. Equipo / Arquitecto del proyecto.
4. Grupo de apoyo.
5. Grupo de clientes.

La pregunta más frecuente en esta etapa son las siguientes:

Q. ¿Cuándo estamos listas para pasar a la siguiente fase?

A. Cuando todas las cuestiones fundamentales del desarrollo, han sido resueltas de manera que todos son preparados para la implementación de los requisitos seleccionados por el cliente.

Otros patrones que forman parte de, están compuestos de asociados estrechamente a la fase de inicialización son:

- El modelo de la fase de exploración, se utiliza para ejecutar la fase de exploración se proporcionan insumos para la misma.

- Este patrón utiliza la configuración del proyecto, la planificación inicial y los patrones de la iteración 0 para producir dicho patrón.
- El patrón de la fase de producción se utiliza para implementar la fase de producción, que el sucesor de la fase de iniciación.

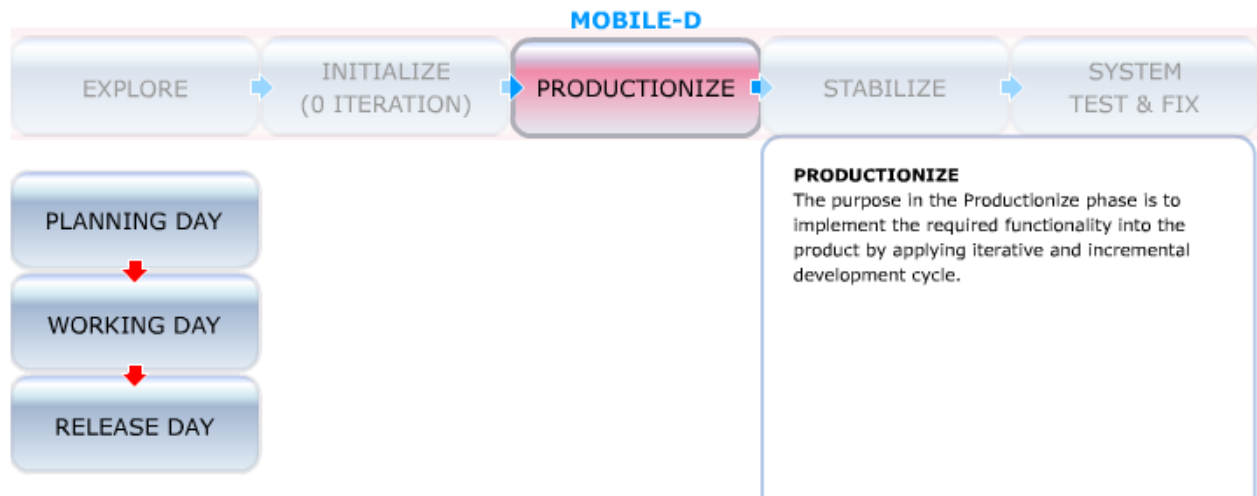
Los posibles riesgos que pueden resultar de la aplicación del patrón de la fase de iniciación, así como sus posibilidades soluciones, que incluyan acciones preventivas para evitar los riesgos y acciones a tomar para minimizar los riesgos y los efectos se discuten aquí:

- Los planes no son flexibles: la solución, haga planes detallados para la siguiente iteración y planes muy aproximados para las próximas fases del proyecto.
- Las cuestiones fundamentales del desarrollo no están listas en el final de la fase: esto puede generar turbulencias en las próximas fases del proyecto, tales como, los retrasos en el calendario de emisiones, estructuras de software inestables y refactorizaciones muy complejas. Así, como una baja productividad y problemas en la calidad del producto final. La solución, de las etapas de la fase de inicialización deben hacerse cuidadosamente. Todos los problemas críticos de desarrollo deben resolverse antes indicando la próxima fase, si es posible, y si no, entonces los riesgos de todas las cuestiones abiertas deben analizarse cuidadosamente y se han iniciado acciones correctivas para minimizar los riesgos. Las partes interesadas proporcionan voluntariamente, todo el apoyo pertinente para que el equipo resuelva, todas las cuestiones críticas del desarrollo abiertas.

1.5.3.Fase de PRODUCCIÓN.

Se usa el desarrollo dirigido por pruebas (TDD), antes de iniciar el desarrollo de una funcionalidad debe existir una prueba que verifique su funcionamiento. En esta fase se lleva a cabo toda la implementación de los módulos, como lo ilustra la figura Figura AP09.

Figura AP9. Ejemplo de la fase de producción.



Fuente: Op. Cit., 2004-2005.

Autor: Juha Koskela y Pakka Kyllönen

Fecha: 07-10-2004

Tipo de patrón: Patrón de fase

El propósito en la fase de producción es implementar la funcionalidad requerida en el producto, mediante la aplicación iterativa y el ciclo de desarrollo incremental.

Los objetivos de la fase de producción son:

1. Implementar la funcionalidad del producto priorizando los requerimientos del cliente, hacia el producto.
2. Enfoque la funcionalidad del núcleo fundamental de su aplicación a principio del incremento para permitir múltiples ciclos de mejora.

Después de la iteración 0:

Los criterios de entrada para la fase de producción son:

1. Se ha completado la fase de la iteración anterior.
2. Se han identificado los requisitos funcionales más importantes.
3. El equipo ha sido ensamblado y ha sido entrenado para el desarrollo.
4. Se ha establecido el entorno de desarrollo.

Las entradas para la fase de producción son:

1. Plan de proyecto actualizado y plan de línea de arquitectura.
2. La primera versión de la arquitectura de software y la descripción del diseño del documento.
3. Planes para el control de cuestiones críticas del desarrollo.
4. Funcionalidad implementada.
5. Cartera de productos, que contiene todos los requisitos identificados del proyecto (tanto funcional como no funcional).
6. Los datos de las métricas.
7. La experiencia del equipo desarrollador del proyecto.
8. Documentación de la prueba de aceptación.
9. Cartas de historias y tareas.
10. Los datos sobre los recursos gastados.
11. Manuales, especificaciones API y otro material de apoyo.
12. Ensayos unitarios.

Después de las siguientes iteraciones:

Los criterios de entrada de la fase de producción son:

1. Completar la iteración de producción.

Las entradas de la fase de producción son:

1. Los resultados de la anterior producen la iteración.

Los criterios de salida de la fase de producción son:

1. La funcionalidad más importante ha sido implementada.

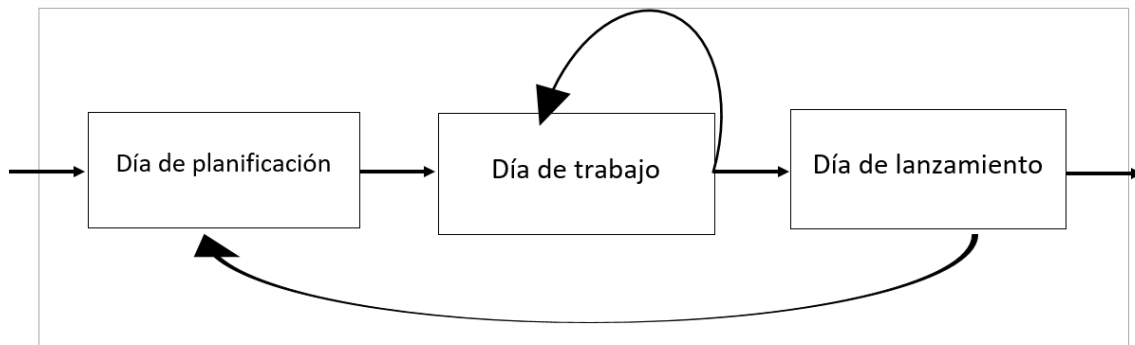
Las salidas de la fase de producción son:

1. La funcionalidad implementada.
2. Documentación de prueba de aceptación.
3. Notas del desarrollador.
4. UI- Ilustraciones.
5. Lista de puntos de acción.

6. Actualización de la pila del producto.
7. Actualización del plan del proyecto.
8. Tarjetas de historias y tareas.
9. Los datos experiencia del equipo desarrollador del proyecto.
10. El conocimiento de los requerimientos del sistema y pruebas de aceptación.
11. Lista de defectos.
12. Divulgación de la lista de verificación para la auditoría.
13. Documentos de requisitos iniciales.
14. Informe de estado diario.

Las etapas de la fase de producción se presentan en la siguiente Figura AP10.

Figura AP10. Etapas de la fase de producción.



Fuente: Op. Cit., 2004-2005.

Las etapas individuales se muestran aquí:

I. Día de planificación.

El propósito en el día de planificación es seleccionar y planificar los contenidos de trabajo para la iteración. Al participar activamente en las actividades de planificación, el cliente garantiza que los requisitos proporcionados, son la mayor parte del valor del negocio, se identifica y los requisitos son correctamente entendido.

- Análisis de requisitos: el propósito de esta tarea es avanzar cuidadosamente en el análisis de los requisitos seleccionados para cada iteración (fase).
- Revisión de pruebas de aceptación: el propósito de esta tarea es asegurarse de que el equipo, entiende los requisitos del sistema correctamente. Esta tarea también permite a los miembros del equipo desarrollador del proyecto, comentando sobre las pruebas de aceptación para mejorar su calidad.
- Generación de pruebas de aceptación: el propósito de esta tarea es verificar los requisitos que el cliente ha fijado para el sistema. Esta tarea actúa como una herramienta de comunicación entre el cliente y el equipo desarrollador del proyecto. Esta tarea incluye una plantilla de aceptación de pruebas.
- Planificación de las iteraciones: el propósito de esta tarea es generar el programa y el contenido para la iteración (fase) a ejecutar. El contenido se define en términos de tareas que son ordenes de trabajo para el equipo. Esta tarea incluye dos plantillas: historias de trabajo y guías de tareas.
- Taller de post-iteración: el propósito de esta tarea, es mejorar iterativamente el proceso de desarrollo del sistema para ajustar mejor a las necesidades del equipo desarrollador, actual del proyecto de software (retroalimentación). Esta tarea incluya una plantilla llamada punto de acción.

II. Día de trabajo

El propósito de esta etapa, es implementar la funcionalidad del sistema planeada durante el día de planificación. El equipo desarrollador se centra en la funcionalidad de mayor prioridad definida por el cliente. Los días del trabajo, se utilizan en la producción, estabilización y fases de prueba y arreglo del sistema. Una iteración puede contener 1-n días laborales. Trabajando los días, forman los días reales del desarrollo de la iteración.

- Programación en parejas: el propósito de la programación en parejas es mejorar la comunicación, mejorando el proceso de la fidelidad y la difusión

del conocimiento dentro del equipo desarrollador del proyecto, y garantizar la calidad del código.

- Integración continua: el propósito es integrar de forma continua con el nuevo código.
- Informes al cliente: el propósito de esta tarea es proporcionar una visión honesta de los avances al cliente, para dar la posibilidad de dar su opinión sobre las características implementadas y para guiar el desarrollo.

III. Día de lanzamiento.

El propósito en esta etapa del desarrollo, se trata de hacer una liberación del sistema completamente funcional.

- Integración del sistema: los productos complejos pueden requerir que el sistema se divida en subsistemas más pequeños. En el caso del proyecto multi-equipos, el propósito de esta tarea es integrar subsistemas, que se generan en equipos, separados, en un solo producto.
- Pruebas de pre- lanzamiento: el propósito de esta tarea es asegurarse de que el software que está siendo producido, está listo para las pruebas de aceptación y liberación.
- Prueba de aceptación: el propósito de esta tarea es verificar que los requisitos, que el cliente ha fijado para el software, se han implementado correctamente.
- Ceremonia de lanzamiento: el propósito de esta tarea es la de confirmar que todo, se ha hecho justo en la iteración actual y siendo la base para un mayor desarrollo, esta garantizada. Las ceremonias de lanzamiento son los pasos finales antes de hacer una nueva versión del software. En la práctica, las ceremonias de lanzamiento consisten en dos actividades esenciales; la auditoría de lanzamiento y la creación de una línea base.

La fase de producción, básicamente tiene los mismos roles de las fases anteriores. Sin embargo, la comunicación con el cliente, se debe enfatizar con retroalimentación rápida

durante la ejecución; es beneficiosa para lograr los resultados satisfactorios. Es decir, mejor correspondencia con los requisitos.

Los siguientes roles están relacionados con la fase de producción:

1. Equipo desarrollador del proyecto.
2. Grupo de soporte / apoyo.
3. Grupo de cliente.
4. Grupo de dirección.

Las respuestas a las preguntas más frecuentes proporcionaran información adicional en profundidad, obtenida al aplicarla en la práctica.

Q. ¿Cuáles son las principales diferencias entre las fases de inicialización, producción y estabilización?

A. El propósito esencial de la fase de inicialización, es permitir que el éxito de las futuras fases del proyecto, mediante la determinación del núcleo de los requerimientos esenciales y familiarizarse con el proceso de desarrollo y las herramientas.

Q. ¿Cuándo estamos listos para pasar a la fase de estabilización?

A. Cuando la funcionalidad más importante del producto (priorizada por el cliente) ha sido completada satisfactoriamente.

Otros patrones que forman parte de, o están compuestos de asociados estrechamente a la fase de producción se identifican aquí:

- Fase de inicialización proporciona, los insumos para esta fase.
- Fase de producción, se compone de la planificación del día, el día de trabajo y liberar los patrones diarios.
- La fase de estabilización continua en esta fase.

1.5.4. Fase de ESTABILIZACIÓN.

En esta fase se realizan las acciones de integración para vincular los posibles módulos separados en una única aplicación, como lo muestra la Figura AP11.

Figura AP11. Ejemplo de la fase de estabilización.



Fuente: Op.Cit., 2004-2005.

Nombre: El patrón de la fase de estabilización.

Autor: Tuomas Ihme.

Fecha: 01-11-2004.

Tipo de patrón: Patrón de fase

Clasificación de patrón: Esencial

El propósito de esta fase de estabilización es asegurar la calidad de la implementación del proyecto, se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funciona correctamente. Esta será la fase más importante de los proyectos multi-equipos, con diferentes subsistemas desarrollados, por equipos de desarrollo distintos. En esta fase los equipos desarrolladores del proyecto realizarán tareas similares a las que debían desplegar en la fase de producción, aunque en este caso el esfuerzo se dirige a la integración del sistema. Adicionalmente se puede considerar en esta fase al inicio de la producción de la documentación.

Los objetivos del patrón de la fase de estabilización son:

1. Finalizar la implementación del producto de software.

2. Mejorar, asegurar y garantizar la calidad del producto.
3. Finalizar con la documentación del producto.

Los criterios de entrada para la fase de estabilización son:

1. Se ha completado satisfactoriamente la fase precedente. Esto significa que la funcionalidad esencial y el valor comercial del proyecto y/o producto, ha sido implementado y se le ha mostrado al cliente.

Las entradas para la fase de estabilización son:

1. La funcionalidad implementada del producto.
2. Los artefactos de desarrollo relacionados.

Los criterios de salida para la fase de estabilización son:

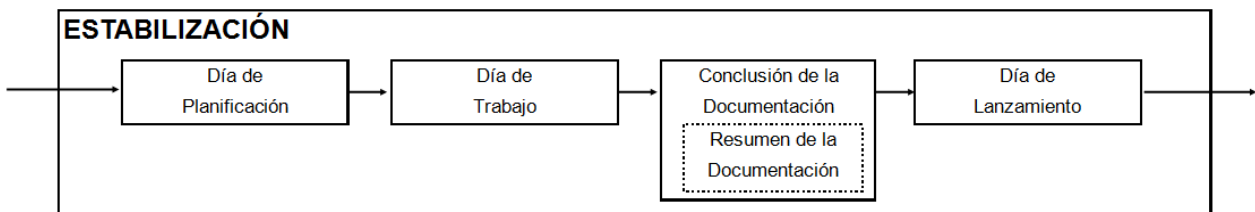
1. El 100% de la funcionalidad necesaria del proyecto y/o producto está funcionando y cumple con las necesidades del cliente.

Las salidas para la fase de estabilización son:

1. Las funcionalidades implementadas de todo el proyecto y de todo el software.
2. La documentación del producto finalizado.

El patrón de la fase de estabilización puede realizarse a través de las siguientes etapas, ilustradas en la Figura AP12:

Figura AP12. Etapas de la fase de estabilización.



Fuente: Op. Cit., 2004-2005.

Las etapas del patrón de estabilización individualmente son las siguientes:

- I. Día de planificación.

El propósito en el día de planificación es seleccionar y planificar los contenidos de trabajo para cada iteración. Al participar activamente en las actividades de planificación, se le garantiza al cliente que los requisitos, que ha proporcionado y que son en su mayoría el valor del negocio, se han identificado plenamente y han sido correctamente entendidos en su totalidad. Al definir el contenido (es decir, historias y tareas), para la aplicación de las demás características del producto y mejorar la calidad externa e interna del producto (refactorización).

II. Día de trabajo.

El propósito de esta etapa es implementar la funcionalidad de la App (sistema), planeada durante el día de planificación. El equipo desarrollador del proyecto se centra en la funcionalidad, definida previamente por el cliente. El día de trabajo, se utiliza en las fases de producción, estabilización y pruebas, además de la fijación del sistema. Una iteración puede contener 1-n días laborales. Los días de trabajo forman, los días reales, de desarrollo de la iteración. En la etapa del día de trabajo de la fase de estabilización, es para finalizar la aplicación del producto, así como mejorar y asegurar la calidad del producto.

III. Conclusión de la documentación.

El propósito de la etapa de conclusión de la documentación es finalizar la arquitectura del software, el diseño de la interfaz de usuario y documentación. Como se muestra en la Figura 12., de las etapas. La etapa de la conclusión de documentación incluye una sola tarea, la tarea de resumen de la documentación. Que puede llevarse a cabo utilizando la documentación, obtenida del patrón de las tareas.

IV. Día de lanzamiento.

El propósito de la etapa del día de lanzamiento es para verificar y validar la funcionalidad implementada y la calidad de todo el software y su documentación. El día de lanzamiento, culmina en la liberación final de todo el software.

Los siguientes roles están relacionados con la fase de estabilización:

1. Equipo desarrollador del proyecto.
2. Equipo desarrollador del proyecto / Arquitecto del proyecto.
3. Grupo de soporte y/o apoyo.
4. Grupo de clientes.
5. Grupo de dirección.

La respuesta a la pregunta más frecuente, proporciona información a profundidad obtenida al aplicarla en la práctica.

Q. ¿Cuáles son las principales diferencias entre la fase de producción y la fase de estabilización?

A. La funcionalidad esencial (valor) y de negocios del proyecto ha sido implementado y mostrado al cliente final, de la anterior fase de producción. El objetivo de la fase de estabilización es finalizar la aplicación, mejorar y garantizar la calidad del producto. Así como finalizar la documentación del producto.

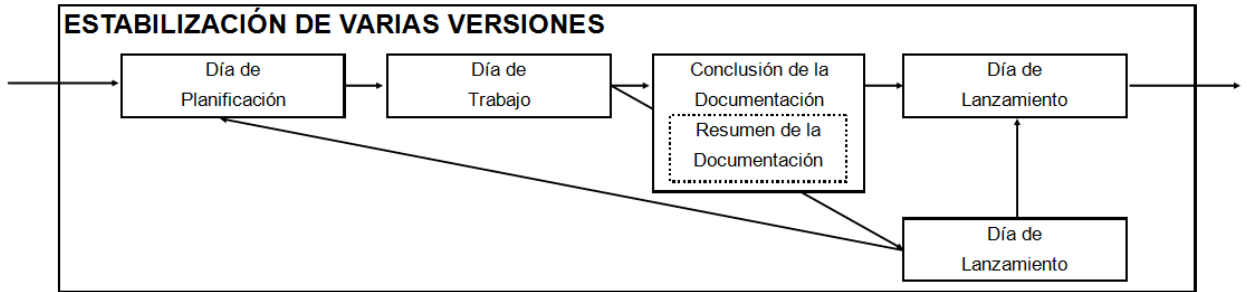
Otros patrones que forman parte de, o están compuestos de asociados estrechamente al patrón de la fase de estabilización:

- El modelo de la fase de producción, se utiliza para implementar la fase de producción, que proporciona insumos para esta fase.
- El patrón utiliza el día de planificación, el día de trabajo, la conclusión de la documentación y el día de lanzamiento, con sus respectivos patrones, para la realización del mismo.
- La carta de ajuste y corrección del sistema, se utiliza para aplicar la prueba del sistema y fijar la fase anterior a la fase de estabilización.

Se ha identificado, la variación siguiente del patrón de etapas de la fase de estabilización, a través de varias tareas de lanzamiento y planificación para finalizar los documentos.

- Estabilizar a través de varias versiones.
Solo se necesita una liberación para ser producida, en una fase de estabilización ideal del patrón, ilustrada en la Figura AP13. Sin embargo, si se necesitan varios lanzamientos, a través de varios patrones de lanzamientos, se puede aplicar. El resumen de la documentación en esta etapa, precede a la versión final solamente.

Figura AP13. Estabilización a través de varios lanzamientos.



Fuente: Op. Cit., 2004-2005.

La variante de la etapa de planificación: tareas de planificación para la finalización de los documentos.

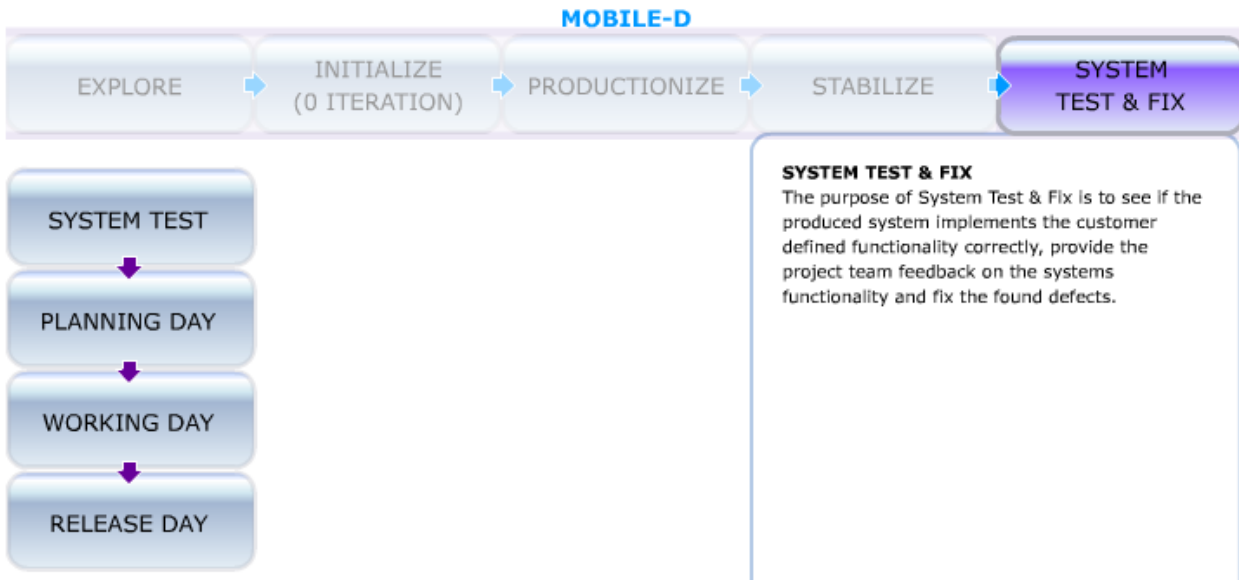
La etapa de planificación de la fase de estabilización, puede incluir la definición del contenido (es decir, historias y tareas), para finalizar la arquitectura del software, diseño y documentos, de la interfaz de usuarios en la etapa de conclusión de la documentación. Los pasos en el patrón de tareas del resumen, de la documentación, son ejemplos, de tareas candidatas.

1.5.5. Fase de PRUEBAS Y CORRECCIÓN DE ERRORES.

Una vez completado el desarrollo del proyecto, se pasa a la fase de pruebas (testeo), hasta llegar a una versión estable del producto, según lo establecido en las primeras fases por el cliente. Sí es necesario se reparan los errores, pero no se desarrolla nada nuevo.

Teniendo en cuenta que una vez acabadas todas las fases, se debió obtener una aplicación publicable y entregable al cliente, como lo muestra la Figura AP14.

Figura AP14. Ejemplos de la fase de pruebas y corrección de errores.



Fuente: Op. Cit., 2004-2005.

Nombre: Prueba y corrección de errores.

Autor: Juho Jääliñoja, Antti Hanhineva y Juha Koskela.

Fecha: 01-11-2004.

Tipo de patrón: Patrón de fase.

Clasificación del patrón: Esencial.

El propósito de la fase de prueba y arreglo, es ver si el sistema producido, implementa correctamente la funcionalidad definida por el cliente, proporcione la retroalimentación de la funcionalidad de los sistemas y corregir los defectos encontrados. Hasta tener una versión estable, según lo establecido anteriormente por el cliente, es decir, como meta para así lograr la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos del cliente.

Los objetivos de la fase de pruebas y corrección de errores del sistema son:

1. Prueba del sistema, basado en la documentación producida en el proyecto y/o producto.
2. Proporcionar información de defectos encontrados.

3. Dejar que el equipo desarrollador del proyecto, para planificar soluciones, para los defectos encontrados.
4. Corregir los defectos, encontrados.
5. Producir como el sistema está libre de errores, como sea posible.

Los criterios de entrada para la fase de prueba y corrección de errores del sistema son:

1. Las fases anteriores necesitan ser terminadas, por lo que las pruebas pueden llevarse a cabo con un sistema en ejecución.
2. Funciones definidas para el equipo de pruebas (testers).

Las entradas para la fase de pruebas y corrección de errores del sistema son:

1. La funcionalidad esta implementada.
2. Aceptación de la documentación de prueba.
3. Funcionalidad definida por el usuario.
4. Descripción de la interfaz de usuario, se utiliza para crear casos de prueba.

Los criterios de salida para la fase de pruebas y corrección de errores del sistema son:

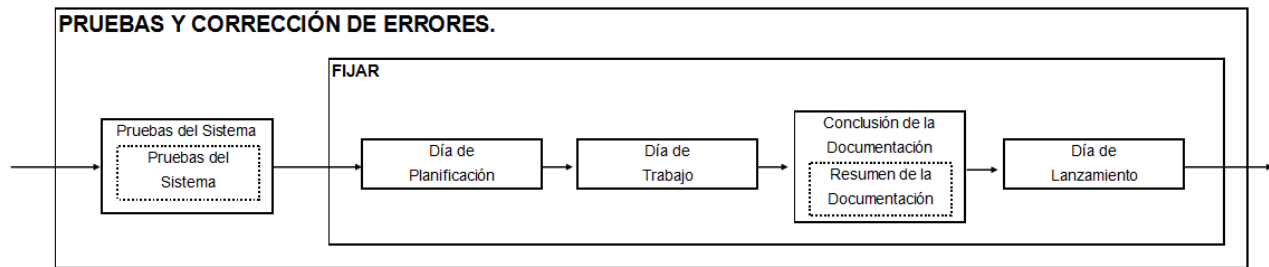
1. Los defectos identificados son fijos.
2. La liberación final, está compuesta como se discute en el patrón relacionado.

Las salidas para la fase de pruebas y corrección de errores del sistema son:

1. Un sistema probado y fijo (la versión final).
2. Documentación de los defectos encontrados.
3. Informe de la prueba del sistema. Una breve descripción del proceso de pruebas y la corrección de los defectos encontrados.
4. Registro de la prueba. Un documento que contiene todos los casos de prueba que serán ejecutados en el ensayo del sistema y resultado de las pruebas.

La siguiente Figura AP15., ilustra las etapas en las que la fase de pruebas y corrección de errores del sistema está compuesta de:

Figura AP15. Etapas de la fase de pruebas y corrección de errores.



Autor: Op. Cit., 2004-2005.

Las etapas individuales de la fase de pruebas y corrección de errores del sistema son:

- La prueba del sistema: es una etapa en la que se prueba el sistema, tal como se describe en el patrón de tareas, de la prueba del sistema. Los defectos encontrados están documentados para el propósito de la iteración “Fijar”.
- Fijar: es una variación de las iteraciones normales. Sin embargo, no hay una nueva funcionalidad y la escala del tiempo, puede ser notablemente más corta. La entrada para esta iteración, son los defectos encontrados, en la etapa de la prueba del sistema.
 - Día de planificación: es el objetivo de la etapa de planificación de la prueba del sistema y arreglos, es definir el contenido (es decir, historias y tareas), para la iteración de fijación. Los defectos encontrados en la etapa de prueba del sistema son entradas para las tareas de descripciones.
 - Día de trabajo: el objetivo de la etapa de trabajo del sistema. La fase de pruebas y fijación consiste en corregir los defectos encontrados en la etapa de prueba y para finalizar la implementación del producto.
 - Conclusión de la documentación: el cliente de la documentación es finalizar la arquitectura del software, el diseño, la interfaz de usuario. La documentación se actualiza para corresponder con los cambios, durante la iteración de fijación. Como se muestra en la Figura 15., en la etapa del resumen de la documentación incluye una sola tarea la documentación, que puede realizarse mediante el uso del patrón de tareas en la conclusión de la documentación.

- Dia de lanzamiento: el propósito de la etapa del día de lanzamiento es verificar y validar la funcionalidad implementada y la calidad en conjunto, software y su documentación. El día de la liberación culmina en la etapa final de todo el software.

Los siguientes roles, están relacionados con la fase de prueba y corrección de errores del sistema son:

1. Equipo desarrollador del proyecto.
2. Grupo de soporte y/o apoyo.
3. Grupo del cliente.
4. Grupo de dirección.
5. Grupo de pruebas (testers) del sistema.

Las respuestas a las preguntas más frecuentes proporcionadas para la fase de pruebas y corrección de errores del sistema, con la información adicional en profundidad, fue obtenida al aplicarla en la práctica:

Q. ¿Cuáles son las principales diferencias entre la fase de producción y la iteración de fijación?

A. En primer lugar, ninguna nueva funcionalidad se implementa durante la solución de la iteración. En segundo lugar, la iteración de fijación suele ser más corta de lo que dura una iteración normal.

Q. ¿Necesitamos un equipo desarrollador del proyecto conjunto, presente durante la iteración del arreglo?

A. No necesariamente. Esto es bastante dependiente de la situación del proyecto. En muchos casos, los defectos encontrados en la prueba del sistema, son bastante sencillos de arreglar y no requieren un equipo entero para participar.

Otros patrones que forman parte de, o están compuestos de asociados estrechamente a las fases de pruebas y corrección de errores del sistema se identifican aquí:

- La fase de estabilización. Proporciona los insumos para esta fase.
- La fase de pruebas y corrección de errores, se compone de la etapa de pruebas del sistema y la iteración de corrección.

- La iteración de fijación se compone del día de planificación, el día de trabajo, la documentación y el día de lanzamiento (Centro de Investigación Técnico VTT de Finlandia, s.f.)