Procesado.

UNIVERSIDAD CATÓLICA ANDRÉS BELLO
VICERRECTORADO ACADÉMICO
ESTUDIOS DE POSTGRADO
ÁREA DE INGENIERÍA
Maestría en Sistemas de Información

Trabajo de Grado de Maestría

MODELO DE MADUREZ PARA LA SELECCIÓN DE UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE PARA LA ADMINISTRACIÓN PÚBLICA NACIONAL

Presentado por:
Castillo Gutiérrez, Luisana

Tutor:
Dr. Pedro Nolasco Bonillo Ramos

Caracas, julio de 2014

AA598/7

UNIVERSIDAD CATÓLICA ANDRÉS BELLO
VICERRECTORADO ACADÉMICO
ESTUDIOS DE POSTGRADO
ÁREA DE INGENIERÍA
Maestría en Sistemas de Información

Trabajo de Grado de Maestría

MODELO DE MADUREZ PARA LA SELECCIÓN DE UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE PARA LA ADMINISTRACIÓN PÚBLICA NACIONAL

Presentado por:
Castillo Gutiérrez, Luisana

Tutor:
Dr. Pedro Nolasco Bonillo Ramos

Caracas, julio de 2014

ACEPTACIÓN DEL ASESOR

Por la presente hago constar que he leído el trabajo de grado de Maestría presentado por la estudiante Luisana Castillo Gutiérrez, titular de la cédula de identidad N° V-12.616.636 para optar al título de Magíster en Sistemas de Información, denominado "Modelo de madurez para la selección de una metodología de desarrollo de software" y manifiesto que cumple con los requisitos exigidos por la Dirección de Postgrado de la Universidad Católica Andrés Bello, y por lo tanto, lo considero apto para ser evaluado por el jurado que se decida designar para tal fin.

En la ciudad de Caracas a los cuatro (4) días del mes de julio de 2014

Atentamente,

Bonillo Ramos Pedro

Dr. Pedro Nolasco Bonillo Ramos
C.I. V-10.868.538

DEDICATORIA

A Dios, por darme la fuerza, el valor y la sabiduría para llegar a este tan esperado momento de hacer mi sueño realidad

A mi hija Louisiana, por ser mi compañera, amiga y apoyo incondicional. Este triunfo es para ti porque eres lo más grande que tengo en la vida. ¡Te adoro!

RECONOCIMIENTOS

A los Profesores de la Universidad Católica Andrés Bello por su ayuda en mi preparación como Magíster en Sistemas de Información.

A mi tutor, el Dr. Pedro Bonillo, por su apoyo en la realización de este proyecto.

Y a todas aquellas personas que de alguna u otra forma colaboraron para que este esfuerzo se hiciera realidad.

A todos, mil gracias por ayudarme a cumplir una nueva meta en mi vida.

ÍNDICE GENERAL

DEDICATORIA ____ *iii*

RECONOCIMIENTOS ____ *iv*

ÍNDICE GENERAL ____ *5*

RESUMEN ____ *13*

INTRODUCCIÓN ____ *14*

CAPÍTULO I ____ *17*

1. EL PROBLEMA DE INVESTIGACIÓN ____ *17*

1.1 Planteamiento del problema ____ **17**

1.2 Formulación de las interrogantes de la investigación ____ **19**

1.3 Objetivos ____ **20**

1.3.1 Objetivo general ____ 20

1.3.2 Objetivos específicos ____ 20

1.4 Justificación ____ **21**

1.5 Alcance ____ **22**

1.6 Limitaciones ____ **22**

CAPÍTULO II ____ *24*

2. MARCO TEÓRICO ____ *24*

2.1 Mapa mental del marco teórico ____ **24**

2.2 Antecedentes de la investigación ____ **25**

2.3 Bases teóricas ____ **27**

2.3.1 Sistemas de información ____ 27

2.3.1.1 Definición de sistemas de información ____ 27

2.3.1.2 Tipos y usos de los sistemas de información ____ 29

2.3.1.3 Importancia de los Sistemas de información ____ 29
2.3.2 Modelos ____ 30
2.3.2.1 Definición de modelo ____ 30
2.3.2.2 Proceso de creación de un modelo ____ 31
2.3.2.3 Clasificación de los modelos ____ 32
2.3.2.3.1 Según su objetivo ____ 32
2.3.2.3.2 Según su nivel de abstracción ____ 32
2.3.2.3.3 Según tipo de representación ____ 33
2.3.2.4 Modelos de madurez ____ 33
2.3.2.4.1 Modelo genérico ____ 34
2.3.3 Metodologías de desarrollo de software ____ 35
2.3.3.1 Definición ____ 35
2.3.3.2 Características ____ 36
2.3.3.3 Clasificación de las metodologías según el modelo de procesos 37
2.3.3.3.1 Modelos convencionales ____ 37
2.3.3.3.1.1 Modelo en cascada ____ 37
2.3.3.3.1.2 Modelo de procesos incrementales ____ 39
2.3.3.3.1.3 Modelo de desarrollo rápido de aplicaciones (DRA) _ 41
2.3.3.3.1.4 Modelos evolutivos ____ 43
2.3.3.3.1.5 MeRinde ____ 45
2.3.3.3.2 Modelos de desarrollo ágiles ____ 47
2.3.3.3.2.1 Programación extrema (XP) ____ 48
2.3.3.3.2.2 Desarrollo adaptativo del software (DAS) ____ 49
2.3.3.3.2.3 Modelo Scrum ____ 50
2.3.3.3.3 Diferencias entre los modelos convencionales y ágiles ___ 52

CAPÍTULO III ____ ***54***

3. MARCO METODOLÓGICO ____ ***54***

3.1 Tipo y Diseño de la investigación ____ **54**

3.2 Población y muestra ____ **55**

3.2.1 Población 55
3.2.2 Muestra 56

3.3 Técnicas e instrumentos de recolección de datos 56
3.3.1 Observación directa 57
3.3.2 Entrevistas 58
3.3.3 Cuestionarios 58
3.3.4 Análisis documental 58

3.4 Medición, validez y confiabilidad 58

3.5 Las variables, sus mediciones e indicadores 60
3.5.1 Definición operacional 61
3.5.2 Definición de variables 61
3.5.3 Operacionalización de las variables 62
3.5.4 Técnicas de procesamiento y análisis de datos 65
3.5.5 Codificación y tabulación de los resultados 66

CAPÍTULO IV 68

4. ANÁLISIS DE LOS DATOS Y PRESENTACIÓN DE RESULTADOS 68

4.1 Resultados en el proceso de investigación de campo 69
4.1.1 Estimación de los tiempos de desarrollo 69
4.1.2 Planificación de las actividades a desarrollar 69
4.1.3 Fecha estimada de finalización del proyecto 70
4.1.4 Cantidad requerida de entregables 70
4.1.5 Tiempo para obtener los primeros resultados 70
4.1.6 Discrepancia entre los resultados y los objetivos 70
4.1.7 Introducir cambios o nuevos requerimientos durante el desarrollo 70
4.1.8 Niveles de re-trabajo 71
4.1.9 Aseguramiento de la calidad del software 71
4.1.10 Nivel de definición de requerimientos 71
4.1.11 Riesgo tecnológico 71

4.1.12 Clientes internos ____ 71
4.1.13 Paradigma de desarrollo ____ 72
4.1.14 Disponibilidad del cliente ____ 72
4.1.15 Impacto financiero ____ 72
4.1.16 Complejidad de la Gerencia de TI ____ 72
4.1.17 Disponibilidad de los servicios de TI ____ 72
4.1.18 Niveles de acuerdo de servicio ____ 73
4.1.19 Apoyo de la alta gerencia ____ 73
4.1.20 Compatibilidad con la metodología de desarrollo ____ 73

4.2 Caso de investigación ____ 73

4.3 Localización ____ 74

4.4 Misión ____ 74

4.5 Visión ____ 74

4.6 Estructura organizacional ____ 74

CAPÍTULO V ____ 76

5. DISEÑO DE LA PROPUESTA ____ 76

5.1 Consideraciones generales ____ 76

5.2 Objetivo de la propuesta ____ 76

5.3 Justificación de la propuesta ____ 76

5.4 Beneficiarios de la propuesta ____ 76

5.5 Evaluación ____ 77

5.6 Estudio de viabilidad o factibilidad de la propuesta ____ 77

5.7 Definición de términos importantes ____ 78

5.7.1 Proceso de software ____ 78
5.7.2 Capacidad de un proceso ____ 78
5.7.3 Madurez de proceso de software ____ 78

5.7.4 Nivel de madurez 79
5.7.5 Áreas clave de proceso 79

5.8 Estructura 79
5.8.1 Niveles de madurez 79
5.8.2 Descripción de los niveles del modelo de madurez 80

5.9 Áreas de proceso 82

5.10 Instrumento de evaluación 83
5.10.1 Escalamiento tipo Likert 83
5.10.2 Codificación de las respuestas 84
5.10.3 Valoración de las preguntas 84
5.10.4 Parámetros de valoración de los niveles de madurez 86

CAPÍTULO VI 87

6. VALIDACIÓN DEL MODELO DE MADUREZ PROPUESTO 87

6.1 Aplicación 87
6.1.1 Niveles de madurez por área de proceso 87
6.1.2 Diagnóstico del grado de madurez 88
6.1.3 Diagnóstico por área 89

CAPÍTULO VII 91

7. CONCLUSIONES Y RECOMENDACIONES 91

7.1 Conclusiones 91

7.2 Recomendaciones 92

REFERENCIAS BIBLIOGRÁFICAS 93

Impresas 93

Web 93

Trabajos de grado 96

FUENTES LEGALES 96

ANEXO A 97

ANEXO B 102

CUESTIONARIO 102

ANEXO C 110

RESULTADOS DEL CUESTIONARIO 110

ANEXO D 110

CONSTANCIA DE VALIDACIÓN DEL CUESTIONARIO 110

ANEXO E 110

ACEPTACIÓN DE LA EMPRESA 110

ÍNDICE DE TABLAS

Tabla **Página**

1. Diferencias entre metodologías ágiles y convencionales o tradicionales 52

2. Definición de variables 61

3. Operacionalización de variables 62

4. Tabulación y escala de las respuestas 66

5. Matriz DOFA de evaluación 77

6. Escala de Likert 84

7. Valorización de prioridades 84

8. Valorización de las preguntas 85

9. Parámetros de valorización 86

10. Niveles de madurez 87

11. Parámetros de valorización 89

12. Respuesta item 1 110

13. Respuesta item 2 ____ *111*
14. Respuesta item 3 ____ *112*
15. Respuesta item 4 ____ *113*
16. Respuesta item 5 ____ *114*
17. Respuesta item 6 ____ *115*
18. Respuesta item 7 ____ *116*
19. Respuesta item 8 ____ *117*
20. Respuesta item 9 ____ *118*
21. Respuesta item 10 ____ *119*
22. Respuesta item 11 ____ *120*
23. Respuesta item 12 ____ *121*
24. Respuesta item 13 ____ *122*
25. Respuesta item 14 ____ *123*
26. Respuesta item 15 ____ *124*
27. Respuesta item 16 ____ *125*
28. Respuesta item 17 ____ *126*
29. Respuesta item 18 ____ *127*
30. Respuesta item 19 ____ *128*
31. Respuesta item 20 ____ *129*
32. Respuesta item 21 ____ *130*
33. Respuesta item 22 ____ *131*
34. Respuesta item 23 ____ *132*
35. Respuesta item 23 ____ *133*
36. Respuesta item 25 ____ *134*
37. Respuesta item 26 ____ *135*
38. Respuesta item 27 ____ *136*
39. Respuesta item 28 ____ *137*
40. Respuesta item 29 ____ *138*
41. Respuesta item 30 ____ *139*

ÍNDICE DE FIGURAS

Figura **Página**

1. Mapa mental del marco teórico 25
2. Sistemas de información 27
3. Modelos 30
4. Modelos de madurez 35
5. Metodologías de desarrollo de software 35
6. Ciclo de vida del modelo en cascada 39
7. Ciclo de vida de modelos incrementales 40
8. Ciclo de vida del modelo de desarrollo rápido de aplicaciones (DRA) 43
9. Ciclo de vida del modelo evolutivo 45
10. Fases de la metodología investigación-acción 55
11. Organigrama del Ministerio del Poder Popular para Industrias 75
12. Niveles de madurez 80
13.Respuestas por área 90

REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD CATÓLICA ANDRÉS
VICERECTORADO ACADÉMICO
ESTUDIO DE POSTGRADO
ÁREA DE INGENIERÍA

MODELO DE MADUREZ PARA LA SELECCIÓN DE UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE PARA LA ADMINISTRACIÓN PÚBLICA NACIONAL

Autor: Luisana Castillo
Tutor: Pedro Bonillo
Fecha: julio, 2014

RESUMEN

El desarrollo de software no es una actividad fácil, muchas veces se utiliza la frase: "codifica y corrige" y sin una metodología definida. El software es diseñado y codificado con los requerimientos mínimos para su funcionamiento. Esto es funcional en desarrollos pequeños sin embargo, a medida que el mismo va creciendo esa forma de hacer las cosas se va tornando cada vez más difícil. Sin contar con los errores que se generan a lo largo del desarrollo. Este estilo de desarrollo se ha usado así por mucho tiempo, sin embargo, existen las metodologías de desarrollo de software, las cuales imponen un proceso disciplinado, entre las que se encuentran: las metodologías tradicionales, las cuales han demostrado ser efectivas y necesarias en un gran número de proyectos pero también han presentado problemas en muchos otros; por otra, las metodologías ágiles, las cuales muestran su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo. El presente trabajo hace énfasis en desarrollar un modelo de madurez para seleccionar una metodología de desarrollo de software en las Oficinas de Gestión de Tecnología y Sistemas de la Administración Pública Nacional. Para lograr ese objetivo, se plantearon cuatro objetivos específicos: Identificación de la problemática existente, determinación de las deficiencias que origina la problemática existente, diseño de las etapas del modelo de madurez y validación del mismo. El ciclo metodológico se basó en la investigación-acción el cual se implementó en cada una de sus cinco fases. El producto final se desarrolló con el objetivo de realizar una evaluación inicial y establecer un proceso de mejora en el proceso de selección de las metodologías de desarrollo.

Palabras clave: Metodologías de desarrollo de software, selección, modelo de madurez, Ingeniería de software, Administración Pública Nacional

INTRODUCCIÓN

La ingeniería de software cuenta con modelos, paradigmas y filosofías de desarrollo, en los cuales se apoya para el desarrollo de sistemas. Entre las filosofías aceptadas se encuentra aquella en donde el proceso de desarrollo es un enfoque bien entendido que se puede planificar, estimar y culminar exitosamente, sin embargo esto en la práctica, no se cumple.

Según el Diccionario de la Real Academia Española (2011), la palabra metodología, se refiere al conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal.

Cuando nos referimos a esta palabra en el área de Ingeniería de Software, según Avison y Fitzgerald (1995) se refiere a "una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información".

Existen múltiples metodologías de desarrollo de software, que coadyuvan, a través de un conjunto de herramientas y prácticas, a obtener un producto de alta calidad y que satisfaga las expectativas del cliente. Se dividen en dos grandes corrientes: tradicionales y ágiles.

Las metodologías tradicionales, hacen énfasis en la planificación de todo el trabajo que se va a realizar y a partir de allí, comienza el ciclo de desarrollo del software. Se enfocan en documentación, planificación y procesos. Han demostrado ser efectivas y necesarias en un gran número de proyectos sobre todo de gran tamaño, sin embargo, también han presentado problemas en muchos otros.

En las metodologías de desarrollo ágil, el desarrollo es iterativo e incremental, los requerimientos y soluciones evolucionan con la colaboración de grupos auto-organizados y multidisciplinarios. Además, la relación con el cliente es primordial debido a la importancia de la colaboración y comunicación entre éstos y el equipo de desarrollo, ya que de esto se deriva el éxito o fracaso del proyecto.

Actualmente, es muy común ver en las empresas el poco conocimiento en cuanto a la selección de una metodología de desarrollo de software que se adapte al proyecto

que se realizará. En ocasiones, se abusa en el uso de una misma metodología de desarrollo de software donde se alcanzan los objetivos del proyecto al precio de un alto sobreesfuerzo personal que asumen los miembros del equipo.

La selección de la metodología es algo muy complicado pues cada proyecto, empresa, etc. requiere de una metodología específica o de diferentes formas de adaptar una metodología genérica.

Es por ello que, ya que se cuentan con esta gran variedad de metodologías y sus particularidades, se desea desarrollar un modelo de madurez que permita facilitar y ayudar en la selección de una metodología de desarrollo de software que mejor se adecúe a las características del proyecto a desarrollar.

En resumen, este trabajo consta de seis capítulos, los cuales se organizan de la siguiente manera:

En el Capítulo I, se presenta la descripción, conceptualización de la problemática que se presentó, el objetivo general y los objetivos específicos. Así como también, la justificación, alcance y limitaciones del problema planteado.

En el Capítulo II, se desarrolla la teoría que fundamenta el proyecto sobre la base del planteamiento del problema que se expuso en el primer capítulo. También, contiene los antecedentes que resultaron más resaltantes y su relación con esta investigación.

En el Capítulo III, se presenta el marco metodológico en el cual se expuso la forma en la cual se realizó el estudio, es decir, el tipo y diseño de la investigación, las técnicas y herramientas utilizadas para el desarrollo del proyecto. Se establecieron los pasos a seguir para realizar el levantamiento de información y el proceso de selección de la población y la muestra. Además, se definieron cuadros de identificación, definición y operacionalización de las variables.

El Capítulo IV, se presentan los resultados obtenidos en la aplicación del instrumento de recolección de datos y el modelo de madurez.

El Capítulo V, se presenta la propuesta de la investigación con los puntos más importantes junto con la justificación y la factibilidad de la misma.

El Capítulo VI, presenta las conclusiones y recomendaciones obtenidas a partir de la problemática existente y la solución propuesta.

Finalmente, se presentan las referencias bibliográficas: web, impresas y trabajos de grado que coadyuvaron en la elaboración de la presente investigación.

CAPÍTULO I

1. EL PROBLEMA DE INVESTIGACIÓN

El objetivo de este capítulo es dar a conocer el planteamiento del problema, los objetivos, justificación, limitaciones del proyecto, con la finalidad de afinar y estructurar formalmente el tema de investigación.

1.1 Planteamiento del problema

Los sistemas de información (SI) han evolucionado y adquirido una importancia estratégica en las organizaciones, de tal forma que ha surgido la necesidad de administrarlos efectiva y eficientemente, a través de una serie de procesos que permiten organizarlos y controlarlos.

Se define metodología como:

> *Una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales puede dividirse en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo Avison y Fitzgerald (1995) (Tinoco y otros, 2010, p.70)*

Hoy en día las necesidades de los clientes son muy cambiantes y las organizaciones deben adaptarse rápidamente a esos cambios, para resolver los problemas y que los proyectos puedan llegar a concluirse de forma exitosa.

Según Advanced Quality Solutions (2002),

> *Aunque la importante evolución de la Ingeniería del Software ha aportado en los últimos años una nueva generación de metodologías y herramientas que hacen que podamos hablar por fin de una industria basada en la ingeniería hoy por hoy el fracaso de proyectos de tecnologías de la información sigue siendo un fenómeno demasiado frecuente, y es que el uso de una metodología adecuada y buenas herramientas son condición necesaria pero no suficiente para el éxito de un proyecto. (p.3)*

En un estudio realizado a la Industria del Software en Venezuela por la Empresa Datanálisis en el año 2003 (como se cita en Méndez, 2006), refleja los principales factores que contribuyen al éxito de las empresas del sector tecnológico, entre las que se destacan la calidad del servicio y el producto entregado; de igual manera hace referencia a las mayores problemáticas que se ven expuestas, entre las que se mencionan: la falta de protección contra la piratería (46,3%), lo reducido del mercado interno (45,0%), la ausencia o carencia de fuentes de financiamiento (43,8%), la falta de una política de estado para la industria del software (31,3%) y el desconocimiento en el ámbito nacional e internacional de los productos y servicios desarrollados en el país (30%). (Méndez, 2006)

Uno de los problemas recurrentes en las organizaciones es que no poseen una metodología formal de desarrollo, si la tienen es aplicada para todos los proyectos, independientemente de la naturaleza del mismo. Tomando en cuenta que ningún proyecto es igual y que cada organización es un mundo, el aplicar una metodología estándar de manera rápida y sin trabajar con los miembros del equipo, suele terminar con bastante frecuencia en un desastre. (Méndez, 2006)

Tener distintas metodologías: tradicionales y ágiles, aplicables de acuerdo con la naturaleza de proyecto, es algo viable. Así se tendría una metodología para cada tipo de proyecto. (Amaro y Valverde, 2007)

Para la selección de una metodología en el desarrollo de software, se debe tener la experiencia suficiente para elegir la que se adapte mejor al proyecto, ya sea tradicional o ágil, que permita maximizar el potencial de los equipos, la calidad del producto con los recursos y tiempos establecidos.

En Venezuela, en el año 2004 y a través de Decreto Presidencial N°. 3390 se ordena a la Administración Pública Nacional el uso prioritario de Software Libre desarrollado con estándares abiertos, en sus sistemas, proyectos y servicios informáticos. Es por ello que el CNTI como parte de la APN y en su rol tecnológico desarrolló la metodología MeRinde (Metodología de la Red Nacional de Integración y Desarrollo de Software Libre) que está basada en varias metodologías como el Proceso Unificado (UP).

Sin embargo, una de las principales desventajas de ésta es que no especifica qué disciplinas, actividades, tareas, roles y subroles se deben emplear conforme a la magnitud del proyecto. El contenido de la metodología es muy amplio y complejo, lo que es visto como muy pesado.

Por otra parte, un estudio realizado por la facultad de Ingeniería de la Universidad de los Andes en conjunto con el Centro de Excelencia en Ingeniería de Software (Ceisoft), el Grupo de Investigación en Ingeniería de Datos y Conocimiento (GIDyC) de la Universidad de Los Andes y el Laboratorio de Investigación en Sistemas de Información (LISI) de La Universidad Simón Bolívar, titulado: "La Industria Nacional del Software en Venezuela", el cual muestra el estado actual de la industria del software en Venezuela (INS), destacando los aspectos de formación y capacitación del personal, como dos de los factores claves para el alcance de los estándares de calidad del software. En el estudio se obtuvieron las siguientes conclusiones:

- El 93% de las empresas utilizan algún modelo de procesos o método de desarrollo en particular. El 51% dice emplear un modelo propio, es decir, han desarrollado su propio modelo de procesos o utilizan una mezcla de varios de los modelos conocidos.
- Sin embargo, se pudo detectar que muchas de las empresas que usan un modelo propio, por lo general, no tienen un modelo documentado a un nivel de detalle que facilite su uso corporativo.

Es por lo anteriormente expuesto y debido a la existencia de una gran cantidad de metodologías que se propone un modelo de madurez para seleccionar, de acuerdo a las características del proyecto a desarrollar, la metodología que mejor se adapte a ésta.

1.2 Formulación de las interrogantes de la investigación

La interrogante principal de la investigación es la siguiente:

¿Cuál es el modelo de madurez para seleccionar la metodología de desarrollo de software que mejor se adecúe al proyecto?

A partir de la anterior interrogante, se derivan los siguientes cuestionamientos:

- ¿Cuál es la problemática existente en relación a selección de la metodología de desarrollo de software que mejor se adapta al proyecto según sus características?
- ¿Cuáles son las deficiencias que originan la problemática existente en la selección de una metodología de desarrollo de software?
- ¿Cuáles son los niveles del modelo de madurez?
- ¿Cómo validar el modelo de madurez obtenido?

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar un modelo de madurez para seleccionar una metodología de desarrollo de software que mejor se adapte a las características del proyecto en la Administración Pública Venezolana.

1.3.2 Objetivos específicos

- Identificar la problemática existente en cuanto a la selección de una metodología para el desarrollo de software.
- Determinar las deficiencias que originan la problemática existente en los proyectos, en cuanto a la selección de una metodología de desarrollo que se adapte a las características del mismo.
- Diseñar los niveles del modelo de madurez, tomando como base los estándares y mejores prácticas existentes.
- Validar el modelo de madurez obtenido, mediante su aplicación en un caso de estudio (Oficina de Tecnología y Sistemas del Ministerio del Poder Popular para Industrias).

1.4 Justificación

Seleccionar una metodología de desarrollo de software que se adapte al tipo de proyecto y la cultura que existe en la empresa es esencial para asegurar la completitud de forma exitosa. Para ello se deben definir desde un principio los pasos a seguir para el desarrollo del mismo, donde sean especificadas las actividades, controles y documentaciones necesarias para el equipo del proyecto y el producto a realizar.

Es por ello, que se desea desarrollar un modelo de madurez para seleccionar una metodología de desarrollo de software más adecuada para llevar a cabo los distintos proyectos que se deseen ejecutar.

Lo que se busca con la selección acertada de la metodología a aplicar en el desarrollo de un software es el aseguramiento de la calidad del producto o servicio a ofrecer, la disminución o mitigación de los riesgos y la facilidad para definir los entregables a realizar.

Es por ello, que el impacto del desarrollo de este trabajo puede ser medido desde diversos puntos de vista, los cuales se detallan a continuación:

- Desde el punto de vista organizacional, porque al contar con varias metodologías de desarrollo de software estaría en la capacidad de aumentar el rendimiento de los equipos de trabajo, mejorar sus procesos para así enfrentar los retos de un entorno dinámico y más exigente.
- En el ámbito económico, debido a que tanto la empresa como el cliente maximizan el retorno de la inversión (ROI) del proyecto.
- A nivel tecnológico, porque así se tendría la capacidad de seleccionar una metodología de desarrollo de software que mejor se adapte a las necesidades del producto a desarrollar, tomando en cuenta las características distintivas de cada proyecto.

1.5 Alcance

Las metodologías a ser utilizadas para el desarrollo del trabajo de grado pertenecen al área de desarrollo de sistemas, las cuales sirven para estructurar, planificar y controlar todo el proceso de desarrollo de un sistema de información. Las metodologías tradicionales a ser utilizadas serán: MeRinde, Modelo en cascada, modelo de procesos incrementales, Modelo de desarrollo rápido de aplicaciones; y por otra parte, las metodologías ágiles: XP, Scrum, Proceso Unificado (UP), FDD.

Como una determinada metodología no es necesariamente aplicable a todo tipo de proyecto, más bien cada tipo de proyecto tiene una metodología a la que se adapta mejor, el modelo que se desarrollará dará como resultado el grado de madurez en el que se encuentra la organización y orientarla hacia un camino ordenado de mejoramiento. Sin embargo, éste no será de uso obligatorio sino que constituirá una guía y orientación para el director y coordinadores de la Oficina de Tecnología del MPPI. Sin embargo, éste no será de uso obligatorio sino que constituirá una guía y orientación para el responsable del proyecto.

Por otra parte, la presente investigación está orientada al contexto académico.

Finalmente, es importante destacar que la aplicación del modelo con proyectos pilotos no está contemplada en ésta investigación.

1.6 Limitaciones

Entre las limitaciones encontradas se tienen:

- La información, del caso de estudio, puede estar limitada debido a que se considere confidencial.
- El modelo a desarrollar no se automatizará a través de una aplicación.
- Las metodologías de desarrollo de sistemas a tomar en consideración en la investigación serán limitadas, ya que, cada una tiene su propio enfoque sobre lo que debería consistir un proyecto de desarrollo de sistemas.
- El modelo de madurez será desarrollado para dos (2) Ministerios de la Administración Pública Venezolana: 1. Ministerio del Poder Popular

para Industrias y 2. Ministerio del Poder Popular para Ciencia, Tecnología e Innovación. Sin embargo, sólo se aplicará en el primero de éstos.

CAPÍTULO II

2. MARCO TEÓRICO

Se hace referencia a los aspectos teóricos que respaldaron la investigación del modelo propuesto. La investigación sitúa los conceptos, bases teóricas, referencias teóricas, conceptos metodológicos y herramientas utilizadas para la propuesta del modelo. Adquiriendo estos conocimientos se realizaron los análisis que tengan lugar durante el estudio para la búsqueda de soluciones al problema planteado.

Según Hurtado (2010), se denomina fundamento teórico al conjunto coherente y coordinado de conceptos, supuestos y proposiciones de carácter explicativo relacionados con la investigación. El marco referencial proporciona al investigador la guía que orienta su estudio para interpretar el resultado de su trabajo, permitiendo su comprensión.

2.1 Mapa mental del marco teórico

En la figura 1, se presenta el mapa mental con los tópicos más importantes relacionados con la teoría que fundamenta la investigación. El tema central es el modelo de madurez para la selección de una metodología de desarrollo de software. En el sentido de izquierda a derecha y de arriba abajo, se esbozan los antecedentes e investigaciones relacionados con la presente, las bases conceptuales y los tres grandes aspectos que detallan la información: los sistemas de información con su definición, tipos y usos e importancia; las metodologías de desarrollo de software con su definición, características, clasificación de las metodologías según el modelo de proceso y según su enfoque; por último, los modelos con su definición, clasificación, tipos de modelos y los modelos de madurez.

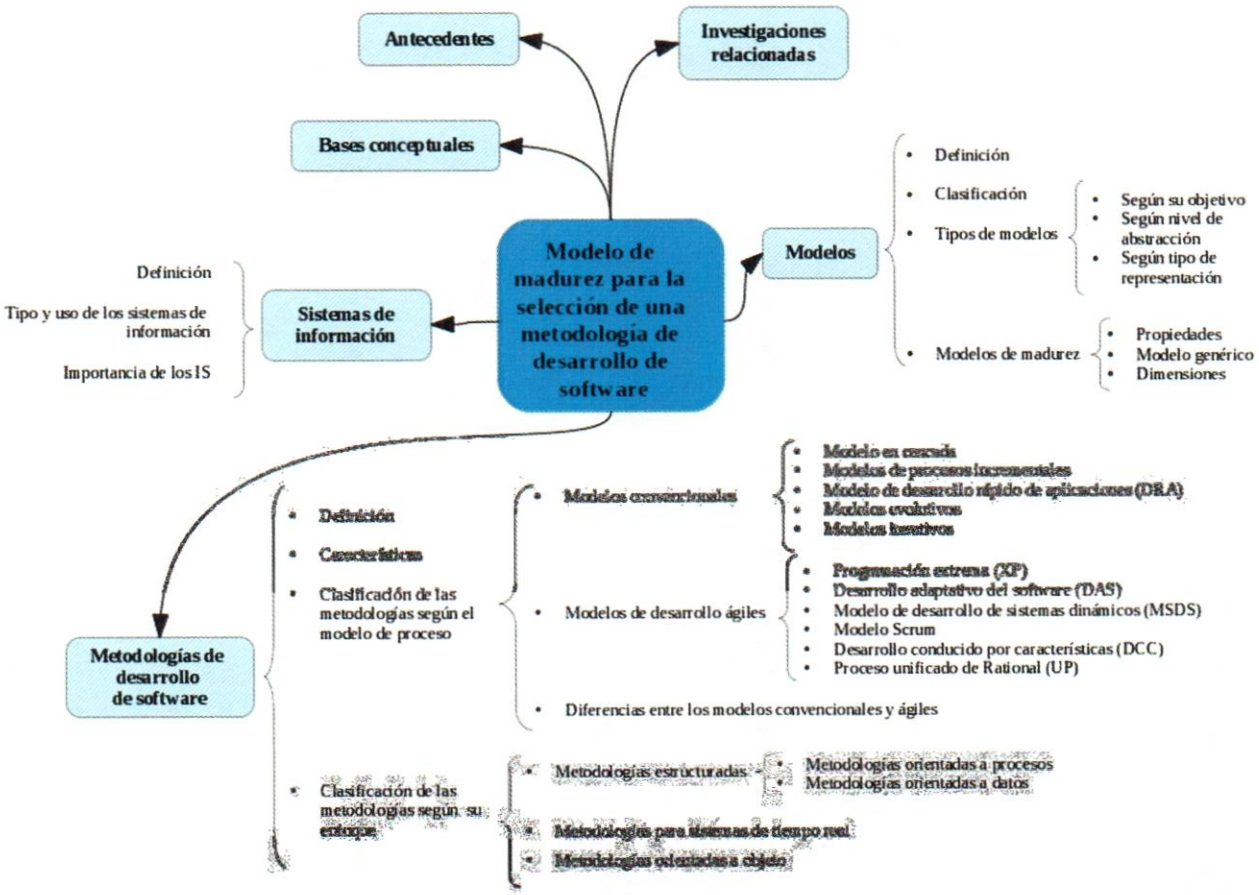


Figura 1. Mapa mental del marco teórico

2.2 Antecedentes de la investigación

Para la realización de este trabajo, se citan los objetivos y resultados obtenidos por otros investigadores y con éstos antecedentes se obtienen conocimientos y se observan las experiencias de otros investigadores que desarrollaron proyectos similares o no al que se plantea en la investigación actual, para ello las principales fuentes estuvieron constituidas por trabajos de grado, referencias electrónicas, artículos, libros cuyo enfoque está orientado a presentar aspectos principales de carácter teórico para la misma. Con esto se puede determinar la factibilidad de proponer un modelo de madurez para la selección de una metodología de desarrollo de software.

Entre las referencias de trabajos de grados de modelos de madurez para la selección de una metodología de desarrollo de software con características similares al objeto de estudio se encontraron:

Angles y Matos (2004), el propósito fundamental de este proyecto fue crear un modelo que asista a la selección de la metodología de desarrollo de software a utilizar para el desarrollo de un proyecto de software, el cual facilitará la toma de decisión al momento de elegir dicha metodología. Con el desarrollo de este proyecto se logró recoger la experiencia de un grupo de especialistas al momento de evaluar y escoger una metodología de desarrollo de software para un proyecto de software; establecer una vinculación directa entre los proyectos y las metodologías de desarrollo de software por medio del modelo, permitiendo demostrar que la selección de una metodología se puede realizar de una manera relativamente objetiva.

El aporte brindado por este trabajo de grado en la presente investigación, fue la orientación obtenida en cuanto a la identificación, definición y operacionalización de las variables, así como también las metodologías de desarrollo de software utilizadas, ya que, para ello, se basaron en la experiencia de especialistas quienes coadyuvaron en la selección de las mismas.

Bastardo (2006), este proyecto fue creado con el objetivo de seleccionar una metodología para la Gerencia de Proyectos de desarrollo de software debido a que existe una gran variedad de metodologías y cada una cuenta con sus particularidades, se diseñó una herramienta que permita facilitar y ayudar a la dirección de proyectos, en la selección de metodologías de desarrollo de software; buscando asegurar que la definición del plan de gestión sea acorde a las características del proyecto a llevar a cabo.

El conocimiento aportado por este trabajo de grado, estuvo relacionado con las características de las metodologías.

Méndez (2006), el proyecto realizado Modelo de evaluación de metodologías para el desarrollo de software, tuvo como objetivo primordial diseñar una matriz de evaluación de metodologías de desarrollo de software. Uno de los aspectos que se resaltan de los resultados obtenidos en esta investigación, es que la matriz de

evaluación de metodologías, la mayor ponderación se presentó en los procesos de desarrollo ágil.

Los conocimientos que éste trabajo de grado proporcionó estuvieron relacionados al marco teórico, específicamente los procesos utilizados en el desarrollo de software, es decir, modelos tradicionales y ágiles.

Vilariño (2010), el objetivo de este trabajo es el desarrollo de un modelo que guíe a un equipo desarrollador en la selección de la metodología más adecuada para el desarrollo de una aplicación Web. Para esto se consideraron algunos aspectos relacionados con el entorno de desarrollo, el producto final, las herramientas disponibles, y la experiencia de los desarrolladores.

Lo aportado por este trabajo de grado para esta investigación fueron los aspectos relevantes a la hora de seleccionar la metodología para llevar a cabo el desarrollo de una aplicación.

2.3 Bases teóricas

Las bases teóricas representan el sustento de la investigación, al recopilar las definiciones, clasificaciones, descripciones y conocimientos aportados por los autores citados.

2.3.1 Sistemas de información

2.3.1.1 Definición de sistemas de información

En la Figura 2, se observa el segmento del mapa mental correspondiente a sistemas de información del marco teórico.

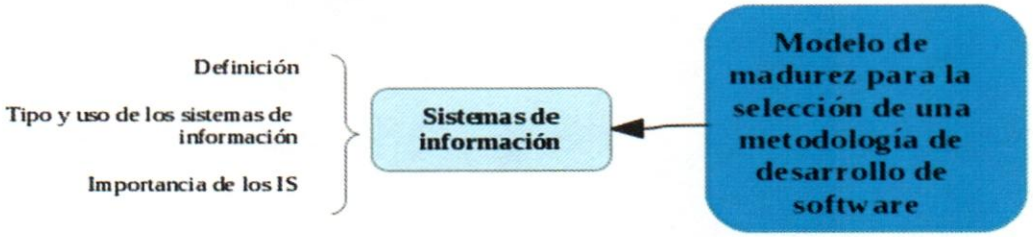


Figura 2. Sistemas de información

Los autores Laudon y Laudon (como se cita en Fernández, 2006), definen los sistemas de información como: "un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de una organización. Además de apoyar en la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores a analizar problemas, visualizar asuntos complejos y crear productos nuevos".

Por otra parte, Andreu, Ricart y Valor (como se cita en Hernández (s/f)), lo definen como: "conjunto formal de procesos que, operando sobre una colección de datos estructurada de acuerdo a las necesidades de la empresa, recopila, elabora y distribuyen selectivamente la información necesaria para la operación de dicha empresa y para las actividades de dirección y control correspondientes, apoyando, al menos en parte, los procesos de toma de decisiones necesarios para desempeñar funciones de negocio de la empresa de acuerdo con su estrategia".

Según O'brien y Marakas (como se cita en Hernández, 2011), un Sistema de Información realiza cuatro actividades básicas:

- Entrada de información: Proceso en el cual el sistema toma los datos que requiere para procesar la información, por medio de estaciones de trabajo, teclado, diskettes, cintas magnéticas, código de barras, etc.
- Almacenamiento de información: Es una de las actividades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardad en la sesión o proceso anterior.
- Procesamiento de la información: Esta característica de los sistemas permite la transformación de los datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general en un año base.

- Salida de información: es la capacidad de un SI para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, graficadores, cintas magnéticas, diskettes, la voz.

2.3.1.2 Tipos y usos de los sistemas de información

Según Muñoz (como se cita en Hernández, 2011), los sistemas de información tienen los siguientes objetivos:

- Automatizar los procesos operativos
- Proporcionar información de apoyo a la toma de decisiones
- Lograr ventajas competitivas a través de su implantación y uso

Con frecuencia, los sistemas de información que logran la automatización de procesos operativos dentro de una organización son llamados Sistemas Transaccionales, en el cual se puede integrar gran cantidad de información institucional para ser utilizada posteriormente por los funcionarios de nivel operativo de la organización en la toma de decisiones.

Por otra parte, los sistemas de información que apoyan el proceso de toma de decisión son los sistemas de apoyo a la toma de decisiones (DSS, por sus siglas en inglés Decisión Support System), generalmente son desarrollados por el usuario final, proporcionan información de soporte para los mandos intermedios y la alta gerencia en el proceso de toma de decisiones. El tercer tipo de sistemas, de acuerdo con su uso u objetivos que cumplen, es de los Sistemas Estratégicos, los cuales se desarrollan en las organizaciones con el fin de lograr las ventajas competitivas, a través del uso de la Tecnología de Información (TI).

2.3.1.3 Importancia de los Sistemas de información

Cuando se habla de la finalidad de un sistema de información, se puede decir que es proporcionar a los ejecutivos, que tienen la potestad de tomar decisiones, de información para confirmar la validez de las estrategias tomadas o a tomar.

Los sistemas deben proporcionar información clave para la toma de decisiones; esta información debe ser sencilla, clara, expedita, veraz, precisa, consistente y fácil de analizar e interpretar. (Fleitman, s/f, p.1)

Es necesario precisar en qué consiste la información clave, cómo puede obtenerse, quiénes son los responsables de elaborarla, en qué formato y con qué frecuencia se va a presentar, quiénes tendrán acceso a ella y con qué detalle y confidencialidad podrá ser consultada. (Fleitman, s/f, p.1)

Es conveniente que la información que generan el sistema de contabilidad y la dirección administrativa se automatice desde que comienzan las operaciones de una empresa. Cuando ésta crece y no se automatiza la información, a veces se vuelve muy complicado pasar de un sistema manual a uno computarizado. (Fleitman, s/f, p.1)

2.3.2 Modelos

2.3.2.1 Definición de modelo

En la Figura 3, se observa el segmento del mapa mental correspondiente a modelos del marco teórico.

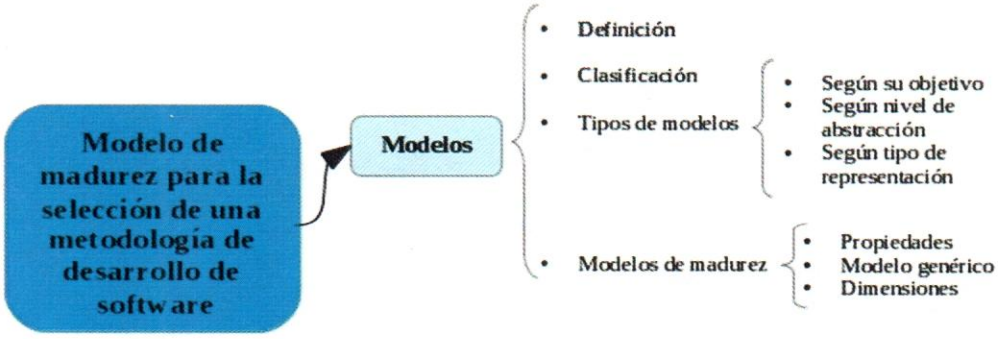


Figura 3. Modelos

El término modelo deriva del latín modellus que significa medida, molde y es asociado a la representación proporcional de alguna cosa. Según señala Hurtado (2010), de esta forma, modelo hace referencia a lo deseado, a un ideal a lo perfecto o lo que debe ser. Así un modelo es la descripción sistemática, consciente y

simplificada de una parte de la realidad, realizada mediante signos, símbolos, formas geométricas o gráficas y palabras.

Los modelos permiten comprender fácilmente la realidad que rodea a un problema dado, como condición indispensable requiere de una abstracción de dicha realidad.

2.3.2.2 Proceso de creación de un modelo

Según Palella y Martins (como se cita en Hernández, 2011), se distinguen en cuatro fases, estas son:

- Definición de la situación problema o ámbito situacional
- Formulación del modelo
- Selección de variables a ser incluidas
- Selección y nivel apropiado de agregación clasificación y definición de los elementos epistemológicos (conocimiento), ontológicos (ser) y axiológicos (valores)
- Decisiones sobre el tratamiento del tiempo
- Especificaciones de los fines u objetivos innovadores para los que se va a utilizar el modelo
- Simulación y/o validación del modelo
- Evaluación del modelo

Se debe tener en cuenta que la cualidad innovadora del modelo, el cual es requisito indispensable para una genuina producción teórica, lo cual está asociado a la idea de producir una innovación que genera mejoras y acarrea cambios.

Entre las ventajas señaladas por Hurtado (2010), relacionada con los modelos se tiene:

- Los modelos muestran los aspectos más relevantes de aquello que representan y facilitan su comprensión
- Permiten establecer nexos entre diversas disciplinas
- Permiten conectar los aspectos teóricos de una situación con sus consecuencias en la realidad

- Hacen posible la confluencia de varias teorías, logrando establecer una visión compleja del aspecto estudiado
- Tienen valor heurístico y permiten el surgimiento de nuevas investigaciones y aplicaciones prácticas del conocimiento.

2.3.2.3 Clasificación de los modelos

2.3.2.3.1 Según su objetivo

- Modelo representativo: Es una representación simplificada de una situación u objeto en base a sus características fundamentales, facilitando así la visualización del objeto en su conjunto.
- Modelo explicativo o teórico: Permite ilustrar la explicación de un proceso o fenómeno, integrando los diferentes eventos que forman parte del mismo y la manera como se relacionan e interactúan.
- Modelo normativo: Proporciona un conjunto de criterios y pautas para la acción, como lineamientos que permiten llevar a cabo cada una de las etapas de un proceso, a modo de plan de acción.
- Modelo operativo: Da cuenta del funcionamiento de ciertos objetos, artefactos o de algún fenómeno en general, visualizando sus procesos de funcionamiento internos.

2.3.2.3.2 Según su nivel de abstracción

- Modelo formal: Está compuesto por un conjunto de conceptos vinculados mediante ciertas relaciones de tipo lógica o analógica, se pueden derivar de las relaciones entre un conjunto de datos de forma inductiva o mediante procesos deductivos. El modelo formal puede estar basado en metáforas o analogías.
- Modelo material: Relaciona características concretas en lugar de conceptos o analogías, se basa en referentes empíricos, representan a un objeto, por lo general son representaciones a escala del objeto físico real.

2.3.2.3.3 Según tipo de representación

- Verbales: Es aquel en el cual se utilizan las descripciones y explicaciones verbales como lenguaje para expresar el fenómeno que se desea representar.
- Gráfico o icónicos: Este tipo de modelos recurre a los gráficos tales como dibujos, imágenes y diagramas, como medio para expresar sus representaciones.
- Matemáticos: Son aquellos que utilizan símbolos numéricos y sus relaciones matemáticas como medio de representación, por lo general toman forma de ecuaciones.

Según las diversas clasificaciones anteriormente dadas, el modelo que propone la presente investigación según su objetivo es de tipo normativo, ya que propone las pautas de selección de una metodología de desarrollo de software. Según su nivel de abstracción es un modelo formal, debido a que está compuesto por conceptos vinculados entre sí relacionados de manera lógica. Según su tipo de representación es un modelo tanto verbal como gráfico, ya que se utilizará descripciones y explicaciones tanto verbales como dibujos, imágenes y diagramas como medio de expresar su representación.

2.3.2.4 Modelos de madurez

El enfoque de los Modelos de Madurez para el control sobre los procesos de TI consiste en desarrollar un método de asignación de puntos para que una organización pueda calificarse desde Inexistente hasta Optimizada (de 0 a 5).

Este planteamiento se basa en el Modelo de Madurez que el Software Engineering Institute definió para la madurez de la capacidad de desarrollo de software. Cualquiera sea el modelo, las escalas no deben estar demasiado simplificadas, lo que haría que el sistema fuera difícil de usar y sugeriría una precisión que no es justificable.

2.3.2.4.1 Modelo genérico

Cobit 4.1 (2007) por el IT Governance Institute, El modelo de madurez consta de seis niveles (ver figura 4):

- No Existente (0). Carencia completa de cualquier proceso reconocible. La empresa no ha reconocido siquiera que existe un problema a resolver.
- Inicial (1). Existe evidencia que la empresa ha reconocido que los problemas existen y requieren ser resueltos. Sin embargo; no existen procesos estándar en su lugar existen enfoques ad hoc que tienden a ser aplicados de forma individual o caso por caso. El enfoque general hacia la administración es desorganizado.
- Repetible (2). Se han desarrollado los procesos hasta el punto en que se siguen procedimientos similares en diferentes áreas que realizan la misma tarea. No hay entrenamiento o comunicación formal de los procedimientos estándar y se deja la responsabilidad al individuo. Existe un alto grado de confianza en el conocimiento de los individuos y, por lo tanto, los errores son muy probables.
- Definido (3). Los procedimientos se han estandarizado y documentado, y se han difundido a través de entrenamiento. Sin embargo, se deja que el individuo decida utilizar estos procesos, y es poco probable que se detecten desviaciones. Los procedimientos en sí no son sofisticados pero formalizan las prácticas existentes.
- Administrado (4). Es posible monitorear y medir el cumplimiento de los procedimientos y tomar medidas cuando los procesos no estén trabajando de forma efectiva. Los procesos están bajo constante mejora y proporcionan buenas prácticas. Se usa la automatización y herramientas de una manera limitada o fragmentada.
- Optimizado (5). Los procesos se han refinado hasta un nivel de mejor práctica, se basan en los resultados de mejoras continuas y en un modelo de madurez con otras empresas. TI se usa de forma integrada para

automatizar el flujo de trabajo, brindando herramientas para mejorar la calidad y la efectividad, haciendo que la empresa se adapte de manera rápida.

Figura 4. Modelos de madurez

Fuente: Cobit 4.1 (2007)

2.3.3 Metodologías de desarrollo de software

2.3.3.1 Definición

En la Figura 5, se observa el segmento del mapa mental correspondiente a metodologías de desarrollo de software.

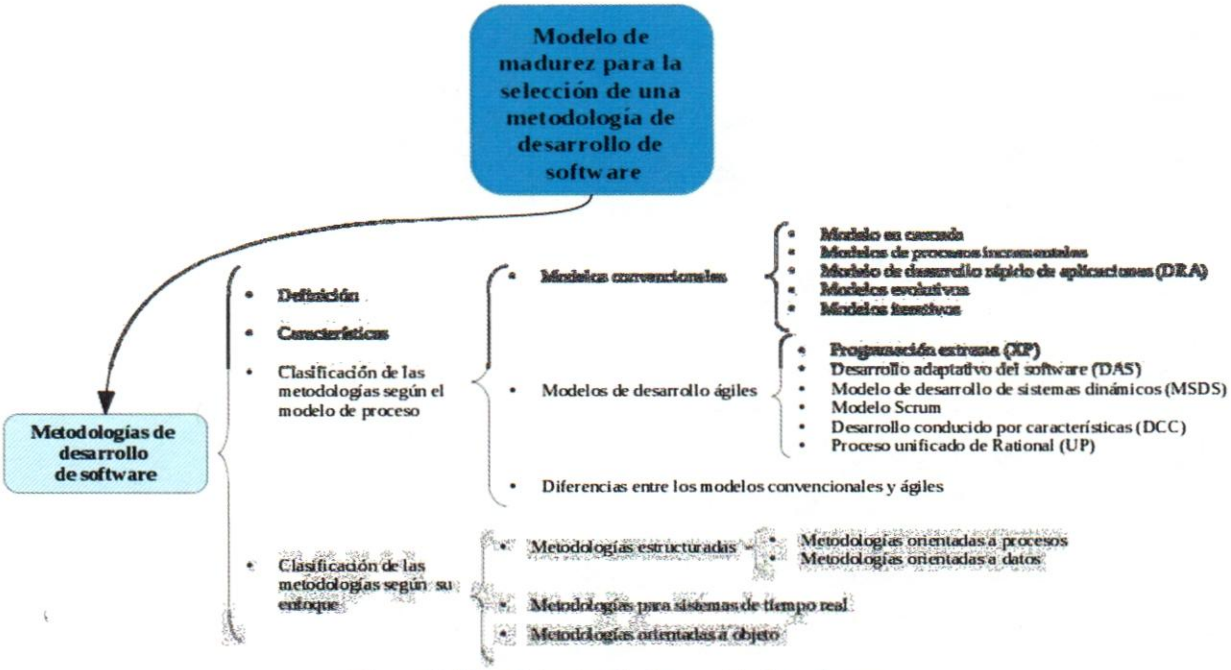


Figura 5. Metodologías de desarrollo de software

Sommerville (2006) define que "un método de ingeniería de software es un enfoque estructurado para el desarrollo de software cuyo propósito es facilitar la producción de software de alta calidad de una forma costeable", cabe destacar que para usar este enfoque se debe manejar conceptos fundamentales tales como; procesos, métodos, tareas, procedimientos, técnicas, herramientas, productos, entre otros.

Específicamente, una metodología se basa en una combinación de los modelos de proceso genéricos para obtener como beneficio un software que solucione un problema. Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades, junto con prácticas, técnicas recomendadas y guías de adaptación de la metodología al proyecto. Sin embargo, la complejidad del proceso de creación de software es netamente dependiente de la naturaleza del proyecto mismo, por lo que la selección de la metodología estará acorde al nivel de aporte del proyecto, ya sea pequeño, mediano o de gran nivel.

2.3.3.2 Características

Lo que se quiere de una metodología es que permita definir las etapas, las salidas, entradas de un proyecto. Mantener un programa no es fácil pero se puede lograr, por lo tanto, las metodologías deben permitir una robusta formación del proyecto que permita utilizar mecanismos de mejora y que se usen los recursos disponibles con su mayor rendimiento y eficacia.

Las características que definen una metodología de desarrollo de software son las siguientes:

- Existencia de reglas predefinidas, fases y subfases, tareas, productos intermedios, técnicas y herramientas tales que se amolden a cualquier desarrollo.
- Cobertura total del ciclo de desarrollo.
- Verificaciones intermedias.
- Planificación y control.
- Comunicación efectiva.

- Utilización sobre un abanico amplio de proyectos.
- Fácil formación.
- Herramientas case.
- Aplicaciones destinadas a aumentar la productividad en el desarrollo.
- La metodología debe contener actividades que mejoren el proceso de desarrollo.
- Soporte al mantenimiento. Por ejemplo. Reingeniería.
- Soporte de la reutilización de software, no solo reutilización de código.
- Actualmente, se huye de métodos muy burocráticos o monolíticos.

2.3.3.3 Clasificación de las metodologías según el modelo de procesos

2.3.3.3.1 Modelos convencionales

Los modelos convencionales surgen para dar orden al caos en los proyectos de desarrollo de software, a través de actividades, acciones, tareas, fundamentos y productos de trabajo que permitirán llevar un mayor control y seguimiento de los mismos. Los modelos convencionales en esencia son más estrictos o rígidos que los de desarrollo ágil, pero nos da mayor seguridad cuando se desea desarrollar un software de alta calidad.

2.3.3.3.1.1 Modelo en cascada

En este tipo de modelo las fases se desarrollan de forma lineal o secuencial. Es ideal cuando se desea realizar adaptaciones o mejoras bien definidas a software ya existentes, en caso de que sea nuevo se hace imprescindible que se tengan los requerimientos bien claros y definidos.

Se denomina así, debido a la cascada de una fase a otra o como ciclo de vida del software. Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo:

- Análisis y definición de requerimientos. Los servicios, restricciones y metas del sistema se definen a partir de las consultas con los usuarios.

Entonces, se definen en detalle y sirven como una especificación del sistema.

- Diseño del sistema y del software. El proceso de diseño del sistema divide los requerimientos en sistemas hardware o software. Establece una arquitectura completa del sistema. El diseño del software identifica y describe las abstracciones fundamentales del sistema software y sus relaciones.
- Implementación y prueba de unidades. Durante esta etapa, el diseño del software se lleva a cabo como un conjunto o unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación.
- Integración y prueba del sistema. Los programas o las unidades individuales de programas se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del software. Después de las pruebas, el sistema software se entrega al cliente.
- Funcionamiento y mantenimiento. Por lo general (aunque no necesariamente), ésta es la fase más larga del ciclo de vida. El sistema se instala y se pone en funcionamiento práctico. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema y resaltar los servicios del sistema una vez que se descubren nuevos requerimientos. (Sommerville, 2006, 62)

Hay ventajas y desventajas en el modelo de cascada. Una ventaja es que cada fase se completa antes de que comience la siguiente. El grupo que trabaja en la fase de diseño, por ejemplo sabe exactamente qué hacer debido a que se ha completado la fase de análisis. La fase de pruebas puede evaluar el paquete completo ya que todo el proyecto está listo.

Una desventaja del modelo de cascada es la dificultad para localizar un problema. Todo el proceso debe investigarse. (Forouzan, 2003, p. 199)

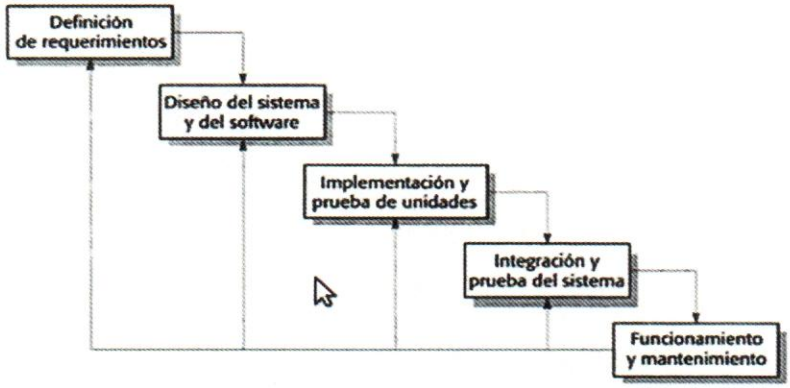


Figura 6. Ciclo de vida del modelo en cascada

Fuente: Forouzan (2003)

2.3.3.3.1.2 Modelo de procesos incrementales

El modelo incremental combina elementos del modelo en cascada aplicado en forma iterativa. El modelo incremental aplica secuencias lineales de manera escalonada conforme avanza el tiempo en el calendario. Cada secuencia lineal produce "incrementos" del software. Por ejemplo, el software procesador de texto, desarrollado con el paradigma incremental en su primer incremento, podría realizar funciones básicas de administración de archivos, edición y producción de documentos; en el segundo incremento, ediciones más sofisticadas, y tendría funciones más complejas de producción de documentos; en el tercer incremento, funciones de corrección ortográfica y gramatical; y en el cuarto, capacidades avanzadas de configuración de página. Se debe tener en cuenta que el flujo del proceso de cualquier incremento puede incorporar el paradigma de construcción de prototipos que se expone más adelante.

A menudo, al utilizar un modelo incremental el primer incremento es un producto esencial. Es decir, se incorporan los requisitos básicos, pero muchas características suplementarias (algunas conocidas, otras no) no se incorporan. El producto esencial queda en manos del cliente (o se somete a una evaluación detallada). Como resultado de la evaluación se desarrolla un plan para el incremento siguiente. El plan afronta la modificación del producto esencial con el fin de satisfacer de mejor manera las necesidades del cliente y la entrega de características y

funcionalidades adicionales. Este proceso se repite después de la entrega de cada incremento mientras no se haya elaborado el producto completo.

El modelo de proceso incremental, al igual que la construcción de prototipos y otros enfoques evolutivos, es iterativo por naturaleza. Pero a diferencia de la construcción de prototipos, el modelo incremental se enfoca en la entrega de un producto operacional con cada incremento. Los primeros incrementos son versiones "incompletas del producto final, pero proporcionan al usuario la funcionalidad que necesita y una plataforma para evaluarlo.

El desarrollo incremental es útil sobre todo cuando el personal necesario para una implementación completa no está disponible. Los primeros incrementos se pueden implementar con menos gente. Si el producto esencial es bien recibido se agrega (si se requiere) más personal para implementar el incremento siguiente. Además, los incrementos se pueden planear para manejar los riesgos técnicos. Por ejemplo, un sistema grande podría requerir la disponibilidad de un hardware nuevo que está en desarrollo y cuya fecha de entrega es incierta. Sería posible planear los primeros incrementos de forma que se evite el uso de este hardware, lo que permitiría la entrega de funcionalidad parcial a los usuarios finales sin retrasos desordenados. (Sommerville, 2006, p. 66)

En el modelo incremental, el proceso se desarrolla en una serie de pasos. El grupo de software primero completa una versión simplificada de todo el paquete. La versión representa el paquete completo pero no incluye los detalles. (Forouzan, 2003, p. 200)

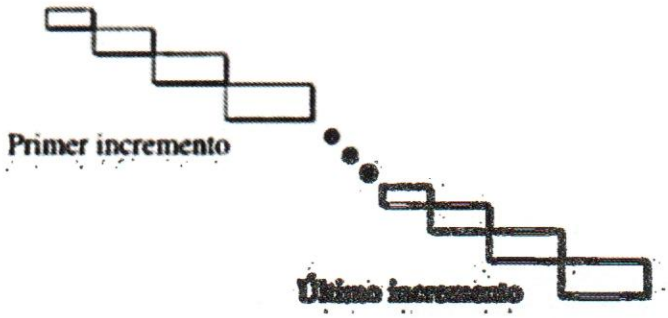


Figura 7. Ciclo de vida de modelos incrementales
Fuente: Forouzan (2003)

2.3.3.3.1.3 Modelo de desarrollo rápido de aplicaciones (DRA)

El desarrollo rápido de aplicaciones (DRA) es un modelo de proceso de software incremental que resalta un ciclo de desarrollo corto. El modelo DRA es una adaptación a alta velocidad del modelo en cascada en el que se logra el desarrollo rápido mediante un enfoque de construcción basado en componentes. Si se entienden bien los requisitos y se limita el ámbito del proyecto, el proceso DRA permite que un equipo de desarrollo cree un sistema completamente funcional dentro de un periodo muy corto (por ejemplo, de 60 a 90 días).

La comunicación trabaja para entender el problema de negocios y las características de información que debe incluir el software. La planeación es esencial porque varios equipos de software trabajan en paralelo sobre diferentes funciones del sistema. El modelado incluye tres grandes fases: modelado de negocios, modelado de datos y modelado del proceso; y establece representaciones del diseño que sirven como base para la actividad de construcción del modelo DRA. La construcción resalta el empleo de componentes de software existente y la aplicación de la generación automática de código. Por último, el despliegue establece una base para las iteraciones subsecuentes, si éstas son necesarias.

Las restricciones de tiempo impuestas sobre un proyecto DRA exigen un ámbito de escalas. Si una aplicación de negocios se puede modular de forma que cada gran función pueda completarse en menos de tres meses (mediante la aplicación del enfoque ya descrito), ésta es una candidata para el DRA. Cada gran función se puede abordar mediante un equipo de DRA por separado, para después integrarlas y formar un todo.

Como todos los modelos de procesos, el enfoque DRA tiene inconvenientes:

- Para proyectos grandes, pero escalables, el DRA necesita suficientes recursos humanos para crear en número correcto de equipos DRA.
- Si los desarrolladores y clientes no se comprometen con las actividades rápidas necesarias para completar el sistema en un marco de tiempo muy breve, los proyectos DRA fallarán.

- Si un sistema no se puede modular en forma apropiada, la construcción de los componentes necesarios para el DRA será problemática.
- Si el alto rendimiento es un aspecto importante, y se alcanzará al convertir interfaces en componentes del sistema, el enfoque DRA podría no funcionar.
- El DRA sería inapropiado cuando los riesgos técnicos son altos (por ejemplo, cuando una aplicación nueva aplica muchas nuevas tecnologías).

Hace un uso intensivo de componentes reusables de software con un ciclo de desarrollo extremadamente corto.

Cuando se utiliza principalmente para aplicaciones de sistemas de información, el enfoque DRA comprende las siguientes fases:

- Modelado de Gestión. El flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la procesa?
- Modelado de datos. El flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.
- Modelado del proceso. Los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos.
- Generación de aplicaciones. El DRA asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso DRA trabaja para volver

a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas para facilitar la construcción del software.

- Pruebas y entrega. Como el proceso DRA enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

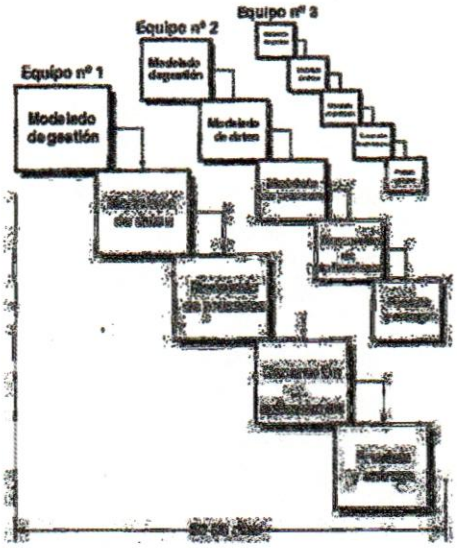


Figura 8. Ciclo de vida del modelo de desarrollo rápido de aplicaciones (DRA)
Fuente: Pressman, 2002

2.3.3.3.1.4 Modelos evolutivos

El desarrollo evolutivo se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado. (Sommerville, 2006, p.78)

Los modelos evolutivos son iterativos, se caracteriza por la forma en que permiten a los ingenieros en software desarrollar versiones cada vez más completas del software. A continuación se presentan algunos de los modelos que se clasifican en esta categoría:

- Construcción de prototipos
- Modelos en espiral
- Modelo de desarrollo concurrente

Existen dos tipos de desarrollo evolutivo:

- Desarrollo exploratorio, donde el objetivo del proceso es trabajar con el cliente para explorar sus requerimientos y entregar un sistema final. El desarrollo empieza con las partes del sistema que se comprenden mejor. El sistema evoluciona agregando nuevos atributos propuestos por el cliente.
- Prototipos desechables, donde el objetivo del proceso de desarrollo evolutivo es comprender los requerimientos del cliente y entonces desarrollar una definición mejorada de los requerimientos para el sistema. El prototipo se centra en experimentar con los requerimientos del cliente que no se comprenden del todo.

En la producción de sistemas, un enfoque evolutivo para el desarrollo de software suele ser más efectivo que el enfoque en cascada, ya que satisface las necesidades inmediatas de los clientes. La ventaja de un proceso del software que se basa en un enfoque evolutivo es que la especificación se puede desarrollar de forma creciente. Tan pronto como los usuarios desarrollen un mejor entendimiento de su problema, éste se puede reflejar en el sistema software.

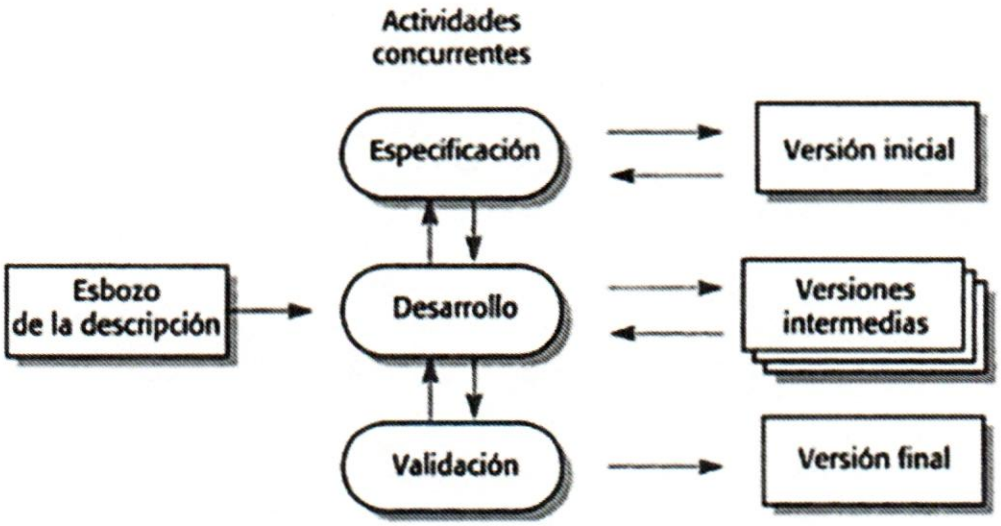


Figura 9. Ciclo de vida del modelo evolutivo
Fuente: Sommerville (2006)

2.3.3.3.1.5 MeRinde

Es un proyecto que propone un estándar abierto para el proceso de desarrollo de software orientado a planes que se estructura en dos dimensiones o ejes. Surge de la combinación y adaptación de modelos y metodologías ampliamente utilizadas para el desarrollo de software y la reingeniería de procesos del negocio. Esta metodología está fuertemente fundamentada en los requerimientos del Centro Nacional de Tecnología de Información (CNTI) y en varias metodologías como el Proceso Unificado (UP) especialmente.

- Fase de inicio. En esta fase se plantea la visión que tiene el equipo o desarrollador en cuanto a lo que será el sistema, se fijan los propósitos o fines principales para el ciclo de vida del producto. Durante la fase de inicio se establece el mecanismo por el cual el producto le proveerá beneficios al usuario final o bien sea al cliente. Se describen todos los actores y casos de usos del producto y además se debe crear o implementar un plan de negocio para definir los recursos que se asignaran al proyecto. Para finalizar esta fase se deben haber tomado en cuenta los costos en recursos, el tiempo total del proyecto, los riesgos e incertidumbres que pueda generar, además de su viabilidad.

- Fase de Elaboración. El propósito específico que tiene la fase de elaboración es proyectar la manera en que se va a realizar la arquitectura para el ciclo de vida del producto, es decir, para su evolución durante su uso o bien sea su permanencia en cuanto a funcionamiento, se elabora una arquitectura en diversas interacciones hasta lograr el producto deseado. Esta fase debe seguir el patrón de todos los casos de uso planteados en la fase de inicio. Además se deben considerar la mayoría de las exigencias funcionales, tomando en cuenta los riesgos que puedan afectar los fines del sistema para que de esta manera pueda hacerse realizable el producto en cuestión. La fase de elaboración concluye cuando el equipo del proyecto tiene en claro los riesgos principales que puedan afectar al producto, de manera de saber cuáles son los requerimientos en cuanto a la realización de este, además de la evolución que este tendrá.
- Fase de Construcción. Una vez que el equipo este en esta fase deben tener como meta o finalidad lograr la disposición o capacidad operativa del producto, considerando que en dicho producto deben de estar incluidas todas las propiedades, elementos, requisitos y/o exigencias, las cuales previamente deben haber sido evaluadas y probadas totalmente, obteniendo de esta manera una versión del producto que sea aprobada o admisible para quien vaya a hacer uso de esta. En conclusión, el objetivo de esta fase es el desarrollo total del sistema ya preparado para la fase de transición, debe haber sido probada toda su funcionalidad y aplicación de manera de evitar que sea pospuesta la fase de transición por incumplimiento de los criterios de esta fase.
- Fase de Transición. Ya en esta fase, el producto debe de estar en manos de los usuarios finales en su forma funcional, luego de que haya sido probado y aceptado en su totalidad por dichos usuarios, además se deberá doctrinar a los usuarios en cuanto al empleo o manipulación del sistema, y principalmente en lo que se refiere a la configuración usabilidad e

instalación del producto. Es decir, se debe avalar o confirmar que el usuario aprenda a operar el producto final, el cual debe cumplir con todos los requerimientos establecidos en el proceso de realización del mismo. En resumen, en esta fase se debe determinar si todos los propósitos en cuanto al proyecto fueron logrados, además se debe confirmar que el cliente haya aceptado, observado y verificado el producto final que le fue proporcionado. (Armadillo Integración Tecnológica, 2013)

2.3.3.3.2 Modelos de desarrollo ágiles

Los modelos de desarrollo ágil no son la antítesis de la práctica sólida de la ingeniería del software, es decir, de la vertiente convencional de los modelos que anteriormente fueron descritos, ya que todos estos modelos se basan de alguna u otra forma en uno o varios de los procesos convencionales, para la definición de sus pasos.

En la actualidad las condiciones del mercado son muy cambiantes, haciendo que las necesidades de los usuarios finales evolucionen, ya que cada día surgen nuevas amenazas competitivas que emergen sin previo aviso. En muchas ocasiones no es posible definir por completo los requisitos antes de que se inicie el proyecto.

Las características esenciales del proceso de desarrollo ágil para la mayoría de los proyectos son las siguientes:

- Dificultad para predecir los requerimientos que persistirán y cuáles cambiarán.
- El diseño y la construcción están intercalados y deben realizarse de manera conjunta de modo que puedan ser aprobados conforme se crean.
- El análisis, el diseño y la construcción no son predecibles.

Motivado a estas características el proceso ágil de software debe adaptarse de manera incremental. La idea es que se mantenga un canal de retroalimentación con el cliente, a través de entregas de prototipos ejecutable o porción de un sistema operacional, en períodos cortos para que la adaptabilidad mantenga un buen ritmo con el cambio.

2.3.3.3.2.1 Programación extrema (XP)

Según Kendall y Kendall (2005), "es un enfoque para el desarrollo de software que utiliza buenas prácticas de desarrollo y las lleva a los extremos". Tiene como objetivo reducir el riesgo en el ciclo de vida del software mediante grupos de desarrollo pequeños, considera que la mejor manera de tratar la falta de requisitos estables en un sistema, es mediante la agilidad de un grupo pequeño de desarrollo.

Esta se basa en la simplicidad, la comunicación y el reciclado continuo de código. El modelo considera varios aspectos problemáticos del desarrollo de software como lo son los retrasos, proyectos cancelados, cambios en el negocio y la rotación del personal.

Sus actividades básicas son: Codificar, hacer pruebas, escuchar y diseñar.

Los objetivos de XP son muy simples:

- La satisfacción del cliente: trata de dar al cliente el software que él necesita y cuando lo necesita.
- Potenciar al máximo el trabajo en grupo: Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

XP define cuatro variables para proyectos de software: coste, tiempo, calidad y ámbito. Además de estas cuatro variables, sólo tres puedan ser establecidas por las fuerzas externas (jefes de proyecto y clientes), mientras que el valor de la cuarta variable debe ser establecido por los programadores en función de las otras tres. (Kendall y Kendall, 2005)

Los valores originales de la programación extrema son:

- Simplicidad: XP propone el principio de hacer la cosa más simple que pueda funcionar, en relación al proceso y la codificación. Esta es la base de la programación extrema.
- Comunicación: Los programadores se comunican constantemente gracias a la programación por parejas y además la comunicación con el cliente es fluida ya que el cliente forma parte del equipo de desarrollo

- Retroalimentación: concreta y frecuente del cliente, del equipo y de los usuarios finales da una mayor oportunidad de dirigir el esfuerzo.
- Coraje o valentía: La valentía le permite a los desarrolladores que se sientan cómodos con reconstruir su código cuando sea necesario. Esto significa revisar el sistema existente y modificarlo si con ello los cambios futuros se implementarán más fácilmente. (Sommerville, 2006)

La programación extrema es uno de los métodos ágiles más conocido y ampliamente utilizados, donde el usuario o cliente forma parte del equipo de trabajo, engloba en una serie de principios, dentro de los más importantes se encuentra la programación en parejas y el desarrollo de pruebas, así como también la reutilización de los códigos. Para el uso de XP los requerimientos deben de estar bien definidos, mediante las historias de usuario. Este modelo se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, especialmente muy utilizada para proyectos con requisitos imprecisos y muy cambiantes, centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo y preocupándose por el aprendizaje de los desarrolladores y la satisfacción del cliente.

2.3.3.3.2.2 Desarrollo adaptativo del software (DAS)

El autor Sommerville (2006), la define como "una metodología para desarrollar el software y sistemas muy complejos. Este se centra en la colaboración humana y la organización del equipo. El Desarrollo adaptativo del software proporciona un marco para el desarrollo iterativo de sistemas grandes y complejos, el mismo fomenta el desarrollo iterativo e incremental con el uso de prototipos".

El ciclo de vida del DAS se conforma de tres fases: Especulación, colaboración y aprendizaje.

- Fase de especulación: Es la primera de las fases, esta inicia en el desarrollo del proyecto. Se utiliza información como la visión del cliente, las restricciones del proyecto y los requisitos básicos para definir el conjunto de ciclos en el que se harán los incrementos del software. En

esta fase es donde se lleva a cabo la planificación tentativa del proyecto en función de las entregas que se irán realizando

- Fase de colaboración: Se busca que el equipo colabore inmensamente para lograr la funcionalidad planeada, se comunique o se encuentre completamente integrados, se desea que exista confianza, donde se puedan realizar críticas constructivas y ayudar si resentimientos, así como también comunicar de una forma oportuna los problemas que se presenten para tomar acciones efectivas y poseer un conjunto de actitudes que contribuyan al trabajo que el equipo de desarrollo se encuentra realizando.
- Fase del aprendizaje: consiste en la revisión de calidad que se realiza al final de cada ciclo, esto permite mejorar el entendimiento real sobre la tecnología, los procesos utilizados y el proyecto. El aprendizaje individual permite al equipo tener mayor posibilidad de éxito.

Esta metodología no proporciona el tipo de prácticas detalladas como lo hace XP, pero proporciona la base fundamental de por qué el desarrollo adaptable es importante y las consecuencias a los más profundos niveles de la organización y la gerencia. Los apoyos filosóficos del DAS se enfocan en la colaboración humana y la organización propia del equipo, y es un modelo para la construcción de software y sistemas complejos, incluye tres fases que son: especulación, colaboración y aprendizaje, cada una de estas fases son fundamental para el desarrollo de la otra. Es adaptativo, se hace uso de la reutilización del código para adaptarlo a lo que se desea.

2.3.3.3.2.3 Modelo Scrum

Scrum, según lo define Citón (2006) "es una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo, fue desarrollado por Jeff Sutherland y elaborado más formalmente por Ken Schwaber. Se enfoca en el hecho de que procesos definidos y repetibles sólo funcionan para atacar problemas definidos y repetibles con gente definida y repetible en ambientes definidos y repetibles. Y se divide un proyecto en iteraciones (que ellos

llaman carreras cortas) de 30 días. La literatura de Scrum se orienta principalmente en la planeación iterativa y el seguimiento del proceso"

Las características son:

- Enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos, prácticas de desarrollo, implementación y demás cuestiones técnicas.
- Hace uso de Equipos auto-dirigidos y auto-organizados.
- Puede ser aplicado teóricamente a cualquier contexto en donde un grupo de gente necesita trabajar junta para lograr una meta común.
- Desarrollo de software iterativos e incrementales basados en prácticas ágiles.
- Iteraciones de treinta días; aunque se pueden realizar con más frecuencia, estas iteraciones, conocidas como Sprint.
- Dentro de cada Sprint se denomina el Scrum Master al Líder de Proyecto quien llevará a cabo la gestión de la iteración.
- Se convocan diariamente un "Scrum Daily Meeting" el cual representa una reunión de avance diaria de no más de 15 minutos con el propósito de tener realimentación sobre las tareas de los recursos y los obstáculos que se presentan. En la cual se responden preguntas como: ¿Qué has hecho desde el último encuentro? ¿Qué obstáculos hay para cumplir la meta? ¿Qué harás antes del próximo encuentro?

El ciclo de vida del modelo Scrum es el siguiente:

- Pre-Juego / Planeamiento: El propósito es establecer la visión, definir expectativas y asegurarse la financiación. Las actividades son la escritura de la visión, el presupuesto, el registro de acumulación o retraso Metodologías Ágiles (backlog) del producto inicial y los ítems estimados, así como la arquitectura de alto nivel, el diseño exploratorio y los prototipos. El registro de acumulación es de alto nivel de abstracción.

- Pre-Juego / Montaje (Staging): El propósito es identificar más requerimientos y priorizar las tareas para la primera iteración. Las actividades son planificación, diseño exploratorio y prototipos.
- Juego / Desarrollo: El propósito es implementar un sistema listo para entrega en una serie de iteraciones de treinta días llamadas "corridas" (sprints). Las actividades son un encuentro de planeamiento de corridas en cada iteración, la definición del registro de acumulación de corridas y los estimados, y encuentros diarios de Scrum.
- Pos-Juego / Liberación: El propósito es el despliegue operacional. Las actividades, documentación, entrenamiento, mercadeo y venta. (Amaro y Valverde, 2007, p. 21)

Scrum es una metodología para la gestión y desarrollo de software basada en un proceso iterativo e incremental, uno de sus principios claves radica en el reconocimiento de que durante un proyecto los clientes pueden cambiar sus pensamientos sobre lo que quieren y necesitan. En este modelo se hacen reuniones diarias o también denominadas reuniones cortas (15 minutos aproximadamente) donde se discute lo que se hizo, lo que se hace, y lo que posteriormente se hará. Es una ayuda para organizar a las personas y el flujo de trabajo, es importante destacar que en este modelo los equipos son auto-organizados (no auto-dirigidos), con margen de decisión suficiente para tomar las decisiones que consideren oportunas.

2.3.3.3.3 Diferencias entre los modelos convencionales y ágiles

Tabla 1. Diferencias entre metodologías ágiles y convencionales o tradicionales
Fuente: Amaro y Valverde (2007)

METODOLOGÍAS ÁGILES	METODOLOGÍAS TRADICIONALES
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparadas para cambios durante el proyecto	Cierta resistencia a los cambios

Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tener metodologías diferentes para aplicar de acuerdo con el proyecto que se desarrolle resulta una idea interesante, estas metodologías pueden involucrar prácticas tanto de metodologías ágiles como de metodologías tradicionales. De esta manera podríamos tener una metodología para cada proyecto, la problemática sería definir cada una de las prácticas, y en el momento preciso definir parámetros para saber cuál usar.

Es importante tener en cuenta que el uso de un método ágil no es para todos. Sin embargo, una de las principales ventajas de los métodos ágiles es su peso inicialmente ligero y por eso las personas que no estén acostumbradas a seguir procesos encuentran estas metodologías bastante agradables.

CAPÍTULO III

3. MARCO METODOLÓGICO

En este capítulo se describe la metodología que permitió desarrollar la presente investigación, se muestran aspectos como el tipo y diseño de la investigación, las técnicas y procedimientos utilizados para su desarrollo. Además, se detallan las variables en estudio y el tratamiento operativo para su evaluación a través de los instrumentos de recolección de datos con la finalidad de obtener la información necesaria para llevar el proyecto a su fase de desarrollo.

3.1 Tipo y Diseño de la investigación

Según lo indica Balestrini (2001), "los proyectos factibles, están orientados a proporcionar respuestas o soluciones a problemas planteados de una determinada realidad: organizacional, social, económica, educativa, etc."

Por lo anteriormente expuesto por Miriam Balestrini y tomando en consideración que el proyecto efectivamente se orienta hacia la solución de un problema planteado, esta investigación es de tipo Proyecto Factible.

Por otra parte, de acuerdo al planteamiento y objetivos que persigue el modelo propuesto se sitúa dentro de la investigación-acción, el cual permite que el investigador pueda intervenir y observar las modificaciones introducidas en el ambiente a estudiar.

Está compuesta por un ciclo de cinco etapas: Diagnóstico, identificación de problema; Planificación de la acción, planificación y fase de investigación para preparar el experimento y acciones alternativas; Acción, aplicación de las nuevas acciones, introducción de cambios y análisis de los resultados del medio ambiente; Evaluación, se determina si se obtienen los resultados esperados o no y asegura que las acciones introducidas son la única razón para el éxito obtenido; Especificación de aprendizaje, identificación de las conclusiones generales.

Figura 10. Fases de la metodología investigación-acción

Asimismo, esta investigación será de campo porque los datos serán tomados de la realidad, ya que no sólo se aplicará la observación sino que se recolectarán los datos directamente de la realidad objeto de estudio, en su ambiente cotidiano, sin manipulación intencional de variables. El autor Arias (1997) define la investigación de campo como: "consiste en la recolección de datos directamente de la realidad donde ocurren los hechos, sin manipular o controlar variable alguna" (p.21).

3.2 Población y muestra

3.2.1 Población

Todo estudio implica la determinación del tamaño poblacional y muestral necesario para su ejecución. Para los autores Palella y Martins (2006), una población de investigación es el conjunto de unidades de las que se desea obtener la información y sobre las que se va a generar conclusiones. La población puede ser definida como un conjunto finito o infinito de elementos, personas o cosas pertinentes a una investigación y que generalmente suele ser inaccesible.

En este trabajo las unidades de análisis objeto de observación o estudio serán las Oficinas de Gestión de Tecnología de Información de dos (2) Ministerios de la Administración Pública Venezolana: 1. Ministerio del Poder Popular para Industrias y 2. Ministerio del Poder Popular para Ciencia, Tecnología e Innovación.

3.2.2 Muestra

Según lo indica Morles (citado por Arias, 1997) la muestra "es un subconjunto representativo de un universo o población".

Debido al que el tamaño de la población es finito y numeroso, se hace necesario estimar una muestra representativa de la misma, para ello existen dos procedimientos para la extracción de la muestra:

- Probabilístico o aleatorio, se aplica si es posible conocer la probabilidad de selección de cada unidad o componente de la muestra.
- No probabilístico, se usa cuando no se puede determinar la probabilidad.

Para esta investigación se utilizó el no probabilístico, junto con el muestreo intencional que según Palella y Martins (2006), ocurre cuando el investigador establece los criterios para seleccionar las unidades de análisis, las cuales reciben el nombre de tipo. Por lo tanto, la muestra para esta investigación viene dada por la Oficina de Tecnología y Sistemas del Ministerio del Poder Popular para Industrias, la cual está conformada por 14 personas.

Esta oficina se utilizará como caso de estudio, para la aplicación y validación del modelo de madurez para la selección de una metodología de desarrollo de software.

3.3 Técnicas e instrumentos de recolección de datos

Las técnicas de recolección de datos, según el autor Arias (1997), son las diversas maneras en las que se pueden obtener la información. Para ello, se utilizan técnicas como la observación, entrevistas, pruebas, encuestas, entre otros.

Ahora bien, según este mismo autor, los instrumentos de recolección de datos, son los medios que se utilizan para recoger y almacenar la información. En estos, se

encuentran: fichas, formatos de cuestionarios, guías de entrevistas, listas de cotejo, escalas de actitudes u opinión (tipo likert), entre otros.

De estos instrumentos, podemos resaltar la entrevista, la cual según Palella y Martins (2006), la define como una técnica que permite obtener datos mediante un diálogo que se realiza entre dos personas. Según estos autores la entrevista es recomendada para investigaciones cuya población o universo es pequeño y manejable, tal como es el caso del presente estudio. Existen dos tipos fundamentales de entrevistas la no estructurada, la cual se caracteriza por la no existencia de estandarización formal, es decir, dejando un margen más o menos grande de libertad para formular las pregunta y proporcionar las respuestas. La entrevista formalizada, por su parte, se basa en un listado fijo de preguntas cuyo orden y redacción permanecen invariables, comúnmente se administra a un gran número de entrevistados para su posterior tratamiento estadístico.

En la presente investigación se utilizó la entrevista no estructurada en su modalidad de entrevista por pautas, que según Palella y Martins (op. cit.), se realizan procurando respuestas a una lista de puntos de interés que se van explorando en el curso de la realización de la entrevista, sobre temas relacionados entre sí.

3.3.1 Observación directa

Mediante ésta técnica de recolección de datos, se recolectarán a través de la propia observación, cómo se realizan los procesos, para así posteriormente ser tomados, analizados y poder determinar los diferentes mecanismos involucrados en la selección de una metodología de desarrollo de software que se adecúe a la naturaleza de los proyectos.

Una vez obtenida la información se definirán los pro y los contra que se encuentren, a fin de lograr el análisis del proceso actual y el enfoque de los problemas que se presentarán.

3.3.2 Entrevistas

Tamayo (2004) expone: "la entrevista es la relación directa establecida entre el investigador y su objeto de estudio a través de individuos o grupos con el fin de obtener testimonios reales", es por ello que, mediante la utilización de éste instrumento de recolección de datos, se efectuarán entrevistas formales e informales a fin de conocer aspectos sobre el flujo de la información. Para ello, se empleará una guía de entrevista y para la recopilación de los datos se utilizará un bloc de notas.

3.3.3 Cuestionarios

Los cuestionarios son herramientas que utiliza el entrevistador para formularle preguntas al respondedor y que éste mismo contesta.

Otra técnica de gran utilidad empleada lo constituirá el cuestionario, el cual, permitirá recoger información a partir de una finalidad previamente establecida. Las preguntas se realizarán de una manera estructurada en base a los objetivos de esta investigación, en un formato de preguntas y respuestas de tipo abiertas y cerradas, obteniendo información general acerca del problema con un orden preciso y lógico introduciendo un plan flexible previamente preparado con relación a las cuestiones que son de interés en el estudio.

3.3.4 Análisis documental

Para la realización del desarrollo de ésta investigación, se consultaron una gran variedad de fuentes de información tales como revistas, informes, trabajos de grado y textos de: metodologías de desarrollo de software, metodologías ágiles, metodologías tradicionales, criterios de selección de metodologías, etc., los cuales se utilizaron para construir las bases teóricas sobre las cuales se estableció el desarrollo del modelo propuesto.

3.4 Medición, validez y confiabilidad

Para el autor Hernández (2010), "la medición es un proceso en el cual se vinculan conceptos abstractos con indicadores empíricos, el cual se realiza mediante

un plan explicito por lo que implica organizar, clasificar y en ocasiones cuantificar la información". Para ello se emplean instrumentos de medición.

Según el autor Hernández (2010), "la validez, estudia la exactitud con que se pueden hacerle mediciones significativas y adecuadas con un instrumento en el sentido que mida realmente el rasgo que pretende medir".

Para ello se utilizará la prueba de expertos, según Balestrini (2001), "se refiere a la revisión exhaustiva del instrumento de investigación antes de ser aplicado"

Por lo tanto, para validar el cuestionarios que se aplicó, se realizó una prueba de expertos que incluyó a un Licenciado en Computación y profesores investigadores del área de Metodología de desarrollo de software, a quienes se les entregó la estructura del mismo con la finalidad de evaluar si el cuestionario recogía en su contenido los aspectos indagados así como también para verificar que la naturaleza de las preguntas que se formulaban sugerían un orden lógico y fácil de seguir para la persona indagada.

Por otra parte, la confiabilidad para Hernández (2010), "es el grado en que un instrumento produce resultados coherentes y consistentes, es decir, en caso de ser aplicados consecutivamente sobre la misma muestra y en un espacio corto de tiempo, el resultado debería ser el mismo". Para Palella y Martins (2006), "la confiabilidad es la ausencia de error aleatorio en el instrumento de recolección de datos, es decir, el grado en que las mediciones están libre de las desviaciones producidas por errores causales".

Para determinar la confiabilidad en la presente investigación, se utilizó el método de coherencia o consistencia interna, el alfa de Cronbach, la cual se aplicó para estimar la confiabilidad del instrumento. Este coeficiente desarrollado por J. L. Cronbach requiere una sola administración del instrumento de medición y produce valores que oscilan entre O y 1. Su ventaja reside en que no es necesario dividir en dos mitades a los ítems del instrumento de medición, simplemente se aplica la medición y se calcula el coeficiente.

Para tal fin se aplicará la siguiente ecuación:

$$rtt = \frac{K}{K-1} * 1 - \frac{\sum Si^2}{St^2}$$

Donde:
rtt = Coeficiente Alpha Cronbach
K = Números de ítemes
$\sum Si^2$ = Sumatoria de la varianza de los puntajes de cada ítem
St^2 = Varianza de los puntajes totales

Fórmula 1. Método de coherencia o consistencia interna

Según Hernández (2010) citado por Hernández, F (2011), otra forma de medir la validez la cual se denomina, validez por expertos, se refiere al grado en que un instrumento de recolección de datos realmente mide la variable que quiere medir de acuerdo al juicio o criterio de personas calificadas, con amplios conocimientos y criterio al respecto.

En esta investigación se utilizó como prueba de validez el juicio de expertos, los cuales validaron el instrumento de recolección de información. (Ver anexo D).

3.5 Las variables, sus mediciones e indicadores

Una variable según Arias (2006) se define como, "una cualidad susceptible de sufrir cambios. Un sistema de variables consiste, por lo tanto, en una serie de características por estudiar, definidas de manera operacional, es decir, en función de sus indicadores o unidades de medida".

El sistema de variables que se presentó en esta investigación, contiene un cuadro donde además de las variables, se especifican sus dimensiones e indicadores y el nivel de medición.

Según Sabino (2007), existen al menos tres tipos de variables, dependiendo de sus relaciones. Las variables independientes, las cuales no se encuentran condicionadas por ninguna otra variable, las variables dependientes, condicionadas

por las variables independientes y las variables intervinientes que actúan con las anteriores.

Cuando las variables son complejas se deben subdividir o descomponer en cualidades más simples y por lo tanto más fáciles de medir, estas constituyen las dimensiones. A su vez, las dimensiones se descomponen en indicadores, que definida por Hurtado (2010), como aspectos específicos y perceptibles de un evento que dan cuenta de la presencia o intensidad del mismo, a modo de indicios.

3.5.1 Definición operacional

Según Palella y Martins (2006), consiste en identificar las variables a estudiar y establecer el significado que el investigador les atribuye dentro de la investigación. De esta manera se identifican los elementos y datos empíricos que expresan y especifican el fenómeno estudiado, indican qué hacer para que el investigador pueda realizar sus observaciones. Así la definición operacional asigna significado a la variable, describiéndola en términos observables y comprobables para poder identificarla, a través de la caracterización o tipificación proporcionados por sus indicadores.

3.5.2 Definición de variables

Tabla 2. Definición de variables

VARIABLE	DEFINICIÓN
Estimación de los tiempos de desarrollo	Es el tiempo estimado en el cual se va a desarrollar el sistema
Cantidad requerida de entregables	Es la cantidad de entregables que se harán durante todo el proceso de desarrollo del sistema
Tiempo para obtener los primeros resultados	Es el tiempo estimado para mostrar al cliente los primeros resultados del desarrollo del sistema
Introducir cambios o nuevos requerimientos durante el desarrollo	Es la flexibilidad de introducir cambios funcionales o de prioridad en cualquier momento durante el desarrollo del sistema
Nivel de definición de requerimientos	Es el nivel de definición que se requiere para los requerimientos iniciales del

	sistema
Riesgo tecnológico	Es la probabilidad ocasione consecuencias en el desarrollo del sistema
Paradigma de desarrollo (lenguaje de desarrollo a utilizar)	Es la propuesta tecnológica adoptada por los desarrolladores o programadores de sistemas
Disponibilidad del cliente	Interacción del cliente con el equipo de desarrollo de sistemas
Impacto financiero	Es el costo asociado al desarrollo o implementación del software
Complejidad de la Gerencia de TI	Es la complejidad en la estructura de la empresa
Apoyo de la alta gerencia	Requerimiento de estudios costo-beneficio para la TI
Resistencia a la innovación de TI	Planes de actualización dentro de la organización para actualizar su infraestructura tecnológica
Capacidad de Adquisición/Desarrollo de TI	Es el tiempo estimado y el tiempo real de entrega del software desarrollado
Gerencia de proyectos	Es la definición de las funciones de los miembros del equipo de desarrollo de software
Compatibilidad con la metodología de desarrollo	Metodología adecuada para el desarrollo de software

3.5.3 Operacionalización de las variables

Tabla 3. Operacionalización de variables

VARIABLE	DIMENSIÓN	INDICADORES
Estimación de los tiempos de desarrollo	• Planificación • Tiempo de entrega • Fecha de entrega	• Largo (Más de 6 meses) • Mediano (Entre 3 y 6 meses) • Corto (Menos de 3 meses)
Cantidad requerida de entregables	• Documentación • Retrasos en los proyectos	• Alta (Documentación de código, notas de versión, instructivo de instalación, ayuda de uso) • Media (Documentación de código y de uso) • Baja (Documentación de código o inexistente)
Tiempo para obtener	• Planificación	• Largo plazo (Más de 6

los primeros resultados	• Medir resultados • Efectividad	meses) • Mediano plazo (Entre 3 y 6 meses) • Corto plazo (Menos de 3 meses)
Introducir cambios o nuevos requerimientos durante el desarrollo	• Ajuste a los objetivos • Re-trabajo • Calidad	• Si • No
Nivel de definición de requerimientos	• Alcance del proyecto	• Específicos • Ambiguos
Riesgo tecnológico	• Equipo de trabajo idóneo • Problemas con los clientes internos	• Alto (Combinación tecnológica no probada) • Medio (Poca experiencia en el uso de la tecnología) • Bajo (Tecnología madura y ampliamente utilizada)
Paradigma de desarrollo (lenguaje de desarrollo a utilizar)	• Aplicable a diversos sistemas • Fácil aprender o enseñar	• Estructurado • Orientados a objeto
Disponibilidad del cliente	• Continuidad de las actividades • Grado de satisfacción	• Alto contacto (Presencia y participación activa) • Contacto moderado (Menor presencia y participación, solo para evaluación) • Bajo contacto (Implica muy poca participación, solo al principio y al final del proceso)
Impacto financiero	• Costo de Desarrollo o Implementación	• 5 = menor que el 25% del valor deseado • 4 = menor que el 50% del valor deseado • 3 = menor que el 75% del valor deseado • 2 = igual al valor deseado • 1 = mayor al valor deseado
Complejidad de la Gerencia de TI	• Complejidad de la estructura	• 5 = Altamente Descentralizados, • 4 = Descentralizados, • 3 = Dispersos, • 2 = Centralizados,

		• 1 = Altamente Centralizados
Complejidad de la Gerencia de TI	• Madurez de los procesos	• 5 = Tiene proceso de TI, • 4 = Tiene y usa los procesos de TI, • 3 = Tiene, usa, documenta los proceso de TI, • 2 = Tiene, usa, documenta, administra los procesos de TI, • 1 = Tiene, usa, documenta, administra y optimiza los procesos de TI
Complejidad de la Gerencia de TI	• Dispersión de los usuarios	• 5 = Altamente Descentralizados, • 4 = Descentralizados, • 3 = Dispersos, • 2 = Centralizados, • 1 = Altamente Centralizados
Complejidad de la Gerencia de TI	• Disponibilidad de los servicios de TI	• 5 = 7x24 díasxhora, • 4 = 7x15 díasxhora, • 3 = 7x12 díasxhora, • 2 = 6x9 díasxhora, • 1 = 5x8 díasxhora
Complejidad de la Gerencia de TI	• Niveles de Acuerdo de Servicio	• 5 = Inmediato o menor a 30 minutos, • 4 = 4 horas, • 3 = Mismo día garantizado, • 2 = Próximo día garantizado, • 1 = Próximo día o mejor esfuerzo
Resistencia a la innovación de TI	• Planes para la renovación periódica de la infraestructura de TI	• 5 = completamente definidas, • 4 = casi todas definidas, • 3 = medianamente definidas, • 2 = poco definidas, • 1 = no están definidas
Apoyo de la alta gerencia	• Requerimiento de estudios costo-beneficio para la TI	• 5 = siempre, • 4 = casi siempre, • 3 = algunas veces, • 2 = pocas veces, • 1 = nunca
Capacidad de Adquisición/Desarrollo de TI	• Tasa de tiempo de respuesta de la organización que adquiere/desarrolla TI	• (Tiempo estimado de entrega / Tiempo real de entrega)

Capacidad de Adquisición/Desarrollo de TI	• Operatividad completa de la TI al ser implantada	• 5 = siempre, • 4 = casi siempre, • 3 = algunas veces, • 2 = pocas veces, • 1 = nunca
Gerencia de proyectos	• Responsabilidad clara de cada miembro del equipo de desarrollo dentro de los proyectos de TI	• 5 = siempre, • 4 = casi siempre, • 3 = algunas veces, • 2 = pocas veces, • 1 = nunca
Compatibilidad con la metodología de desarrollo	• Concordancia de la metodología de desarrollo	• 5 = siempre, • 4 = casi siempre, • 3 = algunas veces, • 2 = pocas veces, • 1 = nunca
Compatibilidad con la metodología de desarrollo	• Cambiar la metodología de desarrollo	• 5 = nunca, • 4 = pocas veces, • 3 = algunas veces, • 2 = casi siempre, • 1 = siempre
Compatibilidad con la metodología de desarrollo	• Previsión del esfuerzo necesario para implementar la TI	• 5 = siempre, • 4 = casi siempre, • 3 = algunas veces, • 2 = pocas veces, • 1 = nunca

3.5.4 Técnicas de procesamiento y análisis de datos

Una vez recolectados los datos mediante las técnicas e instrumentos de recolección de datos que se utilizarán, se codificará la información de forma tal que, la cuantitativa y la cualitativa estén cada una por separado. Con ello se representará la información que se pueda cuantificar y, de la información que se pueda cualificar, se tomará la más relevante.

Para el análisis de la información obtenida en la fase anterior y que fue dividida, se realizará la tabulación de los datos para determinar el número de casos que se ajusten a cada categoría, una serie de gráficos para la información cuantitativa y para la cualitativa se compararán los datos que tuvieran la misma apariencia y se examinarán las diferencias existentes.

De las preguntas abiertas y cerradas practicadas a la muestra se obtendrán unos resultados que serán mostrados posteriormente mediante gráficos y de los resultados obtenidos se determinarán algunos aspectos que formarán parte fundamental para el modelo propuesto.

3.5.5 Codificación y tabulación de los resultados

Para codificar los datos se recurrió al autor Hernández y otros (1998), en el cual se presenta el Diagrama de Likert como técnica de codificación y tabulación de respuestas obtenidas en los instrumentos de recolección de datos.

Según la técnica de Likert, la codificación de respuestas implica cuatro pasos:

- Codificar las categorías de ítems, preguntas y categorías de contenido u observación no precodificada
- Elaborar el libro de códigos
- Efectuar físicamente la codificación
- Grabar y guardar los datos en un archivo permanente.

Según el modelo de la entrevista se tendrán los ítems clasificados según lo contemplado en la siguiente tabla:

Tabla 4. Tabulación y escala de las respuestas

Respuesta	Respuesta	Respuesta	Respuesta	Escala
Muy bajo	Muy poco	Muy escaso	Muy en desacuerdo	1
Bajo	Poco	Escaso	En desacuerdo	2
Medio	Medio	Ni en desacuerdo	Ni de acuerdo	3
Bueno	Frecuente	Alto	De acuerdo	4
Muy bueno	Muy frecuente	Muy alto	Muy de acuerdo	5

Una vez codificadas las respuestas se procederá a establecer las fórmulas y simplificación de los totales obtenidos, para su posterior análisis de tendencias, escalas y fórmulas utilizadas:

Valor mínimo	=>	14 * 1 Punto	=>	14 Puntos
Valor máximo	=>	14 * 5 Punto	=>	70 Puntos
Escala	=>	5 Puntos / Preguntas		

CAPÍTULO IV

4. ANÁLISIS DE LOS DATOS Y PRESENTACIÓN DE RESULTADOS

Según Palella y Martins (2006), la interpretación de los resultados consiste en inferir conclusiones sobre los datos codificados, basándose en las operaciones intelectuales de razonamiento lógico e imaginación, ubicando tales datos en el contexto teórico. La interpretación de los resultados dependen tres factores:

- El nivel de medición de las variables
- La manera como se hayan formulado las hipótesis
- El interés del investigador

Para la realización del modelo de madurez para la selección de una metodología de desarrollo de software se utilizó como metodología principal y de guía, la situada dentro de la investigación-acción.

Siguiendo las pautas de este método, se llevaron a cabo cada uno de los pasos, según se indica a continuación:

- **Obtención de los datos.** Se siguió un plan de elaboración de los cuestionarios y su aplicación en el ente seleccionado para recoger la información.
- **Retroinformación sobre los datos.** Se procesaron los datos recogidos durante la encuesta, lo que conllevó a rediseñar algunas interrogantes del cuestionario elaborado.
- **Análisis de datos.** Se elaboraron las distribuciones de frecuencia para las respuestas obtenidas en la encuesta.
- **Planificación de la acción.** Se inicia con una "idea general" con el propósito de mejorar o cambiar algún aspecto problemático de la práctica profesional. Identificado el problema, se diagnostica y a continuación se plantea la hipótesis acción o acción estratégica. La identificación de la situación problemática se presenta descrita al comienzo de este

documento y se específica como Planteamiento del Problema (Ver planteamiento del problema, Capítulo I, Pág. 12)

- **Acción.** En la investigación acción la reflexión recae principalmente sobre la acción; esto es porque el énfasis se pone en la acción más que en la investigación; la investigación es así mismo revisada, pero su función principal es servir a la acción. Por lo tanto, se desarrolló el modelo de madurez en función de las conclusiones obtenidas en el proceso de recolección de datos.

4.1 Resultados en el proceso de investigación de campo

En el anexo C, se presentan detalladamente los resultados de la problemática actual.

El análisis de los resultados de la investigación con referencia a la encuesta aplicada para establecer el diagnóstico planteado como primer objetivo específico, arrojó los siguientes resultados:

4.1.1 Estimación de los tiempos de desarrollo

Los resultados obtenidos en la investigación, con respecto a los procesos de estimación de tiempo y planificación de las actividades a desarrollar dentro del proyecto, resultaron favorables ya que, se estiman los tiempos del producto a desarrollar, se planifica el desarrollo y posteriormente se ejecuta el plan. En la mayoría de los desarrollos se utiliza la experiencia adquirida en proyectos anteriores.

4.1.2 Planificación de las actividades a desarrollar

Según los encuestados, resulta favorable la forma en la que se establecen las tareas a ejecutar dentro del proyecto, utilizando cronogramas con herramientas de planificación.

4.1.3 Fecha estimada de finalización del proyecto

Los proyectos se programan a partir de una fecha de comienzo conocida, aun cuando se conoce la fecha en la que debe finalizar el proyecto.

4.1.4 Cantidad requerida de entregables

Documentación entregada

El 57% de los encuestados considera que es regular o mala, debido a que la mayoría de los proyectos no tienen ni siquiera un documento que incorpore como mínimo, definición de los objetivos del sistema, justificación técnica, económica y financiera del proyecto, documentación de datos, archivos y bases de datos, etc.

4.1.5 Tiempo para obtener los primeros resultados

Medición de los resultados contra los planes

Cuando se determina el estado del proyecto respecto a los planes, se generan alertas sobre posibles desviaciones para su corrección a tiempo. Sin embargo, el 50% de los encuestados opina que esto no ocurre, lo que conlleva a la improvisación.

4.1.6 Discrepancia entre los resultados y los objetivos

Se toman a tiempo acciones correctivas, si se detecta que existen discrepancias entre las estimaciones respecto de la realidad y se corrigen las desviaciones para disminuir los riesgos futuros.

4.1.7 Introducir cambios o nuevos requerimientos durante el desarrollo

Procesos de control de cambios

El 50% de los encuestados considera que no se definen procesos de control de cambios en los proyectos, lo cual incide negativamente en la culminación con éxito del mismo.

4.1.8 Niveles de re-trabajo

No se hace de forma adecuada la gestión de los requisitos por lo tanto es necesario hacer correcciones para resolver defectos o los requisitos establecidos, muchas veces debido a una mala comprensión de los mismos.

4.1.9 Aseguramiento de la calidad del software

No existe una evaluación continua de la calidad del software, más allá de la que realiza el desarrollador, para asegurar que se cumple con los requisitos especificados, tanto técnicos como funcionales. Esto debido a que no se cuenta con el personal calificado para cubrir esta actividad.

4.1.10 Nivel de definición de requerimientos

Alcance del proyecto

No se tiene claridad en cuanto a la descripción del trabajo que se requiere para entregar el proyecto, por lo tanto el 64% de los encuestados considera que es ambigua esta definición.

4.1.11 Riesgo tecnológico

Combinación entre riesgo tecnológico y personal idóneo

Se tiene poca experiencia en el uso de la tecnología debido a la alta rotación de personal para el desarrollo, el cual en muchas ocasiones tiene poca experiencia en el uso de la herramienta tecnológica que debe ser utilizada para el desarrollo de los proyectos.

4.1.12 Clientes internos

Es considerable la cantidad de problemas que se generan con los clientes internos, debido a la falta de definición clara de los objetivos, alcance de los proyectos y una comunicación eficaz.

4.1.13 Paradigma de desarrollo

Metodología utilizada

El 100% de los encuestados utilizan una metodología de desarrollo de software orientada a objetos, para todos los proyectos que se elaboran, no discriminando para ello las características de los mismos.

4.1.14 Disponibilidad del cliente

Acciones del cliente

Los clientes mantienen contacto moderado para el seguimiento de las actividades de los proyectos.

4.1.15 Impacto financiero

Costos del desarrollo

El 57% de los encuestados considera que los costos del proyecto son mayores o iguales al valor deseado, debido a que se generan problemas como re-trabajo, rotación de personal, cambio en la definición de las actividades.

4.1.16 Complejidad de la Gerencia de TI

Desarrollo de los procesos de negocio

El 64% de los encuestados opina que es insuficiente el desarrollo de los procesos de negocio dentro de la organización, aun cuando se tienen en muy pocos casos son usados, documentados, administrados y optimizados, lo cual permitiría, operaciones simplificadas y de bajo costo. Además, la habilidad de comunicar de forma inmediata información importante a los altos directivos y mejoraría la productividad dentro de la organización.

4.1.17 Disponibilidad de los servicios de TI

Los servicios de TI se encuentran disponibles los 7 días de la semana, las 24 horas del día y funcionan correctamente siempre que los clientes y usuarios desean hacer uso de ellos.

4.1.18 Niveles de acuerdo de servicio

El nivel de calidad del servicio, se garantiza en mayor medida para el mismo día, lo que genera un gran compromiso entre la organización con el servicio prestado a sus clientes o usuarios.

4.1.19 Apoyo de la alta gerencia

Estudios de costo-beneficio

No se realizan la medición de los costos que se incurren en el desarrollo de los proyectos ni se comparan los costos con los beneficios esperados.

4.1.20 Compatibilidad con la metodología de desarrollo

Concordancia entre metodología de desarrollo y proyectos

La metodología de desarrollo utilizada no se adapta al proyecto a elaborar. Se abusa en el uso de una misma metodología donde se alcanzan los objetivos del proyecto al precio de un alto sobreesfuerzo que es asumido por los integrantes del equipo de desarrollo.

4.2 Caso de investigación

Aquí se presenta la aplicación de la propuesta metodológica a un caso real, con la finalidad de poner en práctica la propuesta metodológica, determinando así si el modelo se debe refinar y probar su utilidad.

Por otra parte, se seleccionó al Ministerio del Poder Popular para Industrias. El criterio utilizado para seleccionar a este ente gubernamental como objeto de estudio, está motivado a la facilidad de acceso de la autora, para obtener información de los procesos manejados allí.

El Ministerio del Poder Popular para Industrias, nace con el supremo compromiso de lograr la mayor eficacia política y calidad revolucionaria en la construcción y consolidación del nuevo modelo productivo socialista, basado en el proceso de transformación de la economía rentística e importadora en la economía

productiva y diversificada, potenciando la capacidad interna de producción de bienes y servicios.

Las competencias están orientadas fundamentalmente al desarrollo de las industrias básicas y ligeras, pequeñas y medianas industrias, con ubicación adecuada en el territorio nacional, así como cualquier otra forma asociativa de carácter mixta o privada de bienes de capital e intermedios.

4.3 Localización

El área administrativa de la organización se encuentra en la ciudad de Caracas, específicamente en la Avenida Urdaneta.

4.4 Misión

Ejercer la rectoría del Ejecutivo Nacional, en el desarrollo de las industrias básicas, intermedias y ligeras del país, mediante la inclusión social en el proceso productivo, promoviendo la transferencia tecnológica y la innovación de tecnología nacional, para la instalación, re-activación, re-conversión y re-impulso de la industrias nacional, hacia la satisfacción de las necesidades nacionales y locales de la población, sustituyendo las importaciones, en el marco de los valores y principios socialistas, que conlleven al fortalecimiento de la economía productiva.

4.5 Visión

Ser reconocido como la institución del Estado que vela por el nuevo modelo de desarrollo productivo integral de la industria nacional, ejecutando políticas y acciones orientadas a la instalación y consolidación de industrias, qué contribuyan activamente en la mejora de la calidad de vida de los venezolanos y venezolanas, la diversificación y soberanía productiva de la nación, bajo un esquema socialista basado en la distribución equitativa de la producción, que satisfaga las necesidades de la población.

4.6 Estructura organizacional

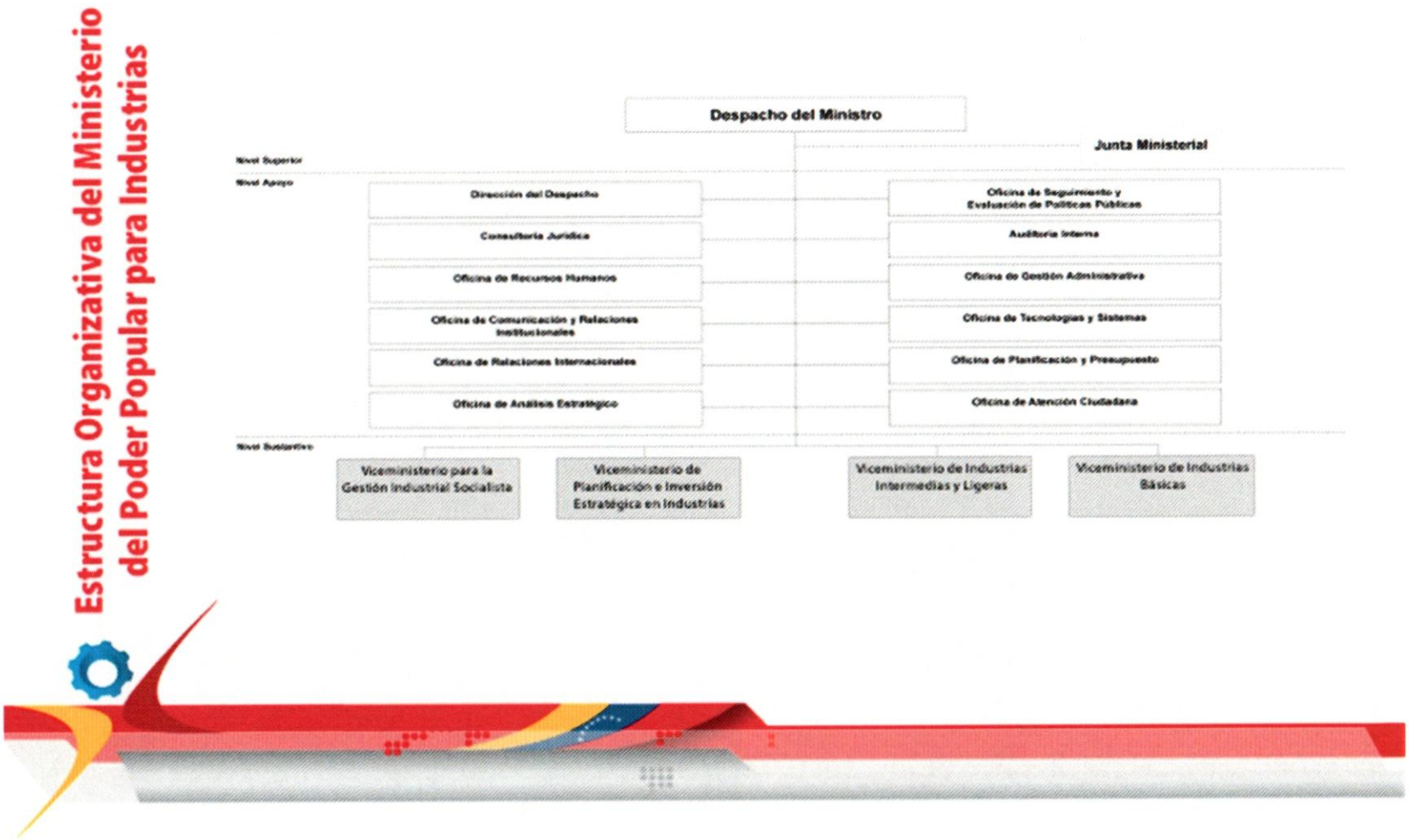


Figura 11. Organigrama del Ministerio del Poder Popular para Industrias

CAPÍTULO V

5. DISEÑO DE LA PROPUESTA

El problema fundamental de las Oficinas de Gestión de Tecnología y Sistemas es la inhabilidad para administrar sus procesos. El modelo de madurez para la selección de una metodología de desarrollo de software se convierte en una guía que coadyuva en el control sobre estos procesos. La meta a alcanzar va a ser la evolución a través de estrategias de mejora determinando la madurez del proceso actual e identificando los puntos importantes que se deben atacar.

5.1 Consideraciones generales

Este capítulo contiene la propuesta de la investigación con los puntos más importantes junto con la justificación y la factibilidad de la propuesta.

5.2 Objetivo de la propuesta

Presentar un modelo de madurez para la selección de una metodología de desarrollo de software para la Administración Pública Nacional.

5.3 Justificación de la propuesta

El modelo de madurez para la selección de una metodología de desarrollo de software beneficiará a las Oficinas de Tecnología y Sistemas de la Administración Pública Nacional, ya que a través de éste se podrá tener una mayor conocimiento de las potencialidades y debilidades de este proceso, de tal forma que se puedan emprender acciones de mejoramiento priorizadas y con mayor probabilidad de éxito.

5.4 Beneficiarios de la propuesta

Esta propuesta beneficiará directamente a las Oficinas de Tecnología y Sistemas de la Administración Pública Nacional (particularmente a los Ministerios del Poder Popular). Además, se beneficiarán indirectamente clientes y usuarios de los proyectos desarrollados.

5.5 Evaluación

La evaluación de la propuesta se realizó a través de la matriz DOFA, la cual es una herramienta utilizada como método de diagnóstico en el ámbito de la planificación estratégica. (Ver tabla 5)

Tabla 5. Matriz DOFA de evaluación

FORTALEZAS	DEBILIDADES
• Modelo de madurez elaborado en función de las necesidades de las Oficinas de Tecnología y Sistemas de la Administración Pública Nacional • Ahorros de tiempos y costos en la elaboración de proyectos • Mejora de la imagen y de la confianza de los clientes internos y externos hacia las Oficinas de Tecnología y Sistemas de la Administración Pública Nacional	• Concienciar al personal sobre la importancia de selección la metodología de desarrollo de software que mejor se adecúe al proyecto • Imposibilidad de prever los aspectos que pueden causar desvíos en los proyectos
OPORTUNIDADES	**AMENAZAS**
• Capacidad de adaptación a las metodologías que se establezcan • Mejora en los proyectos de software que se desarrollen, lo cual conllevará a alcanzar las metas establecidas	• Poca formación del personal de liderazgo de proyectos

5.6 Estudio de viabilidad o factibilidad de la propuesta

La factibilidad y puesta en marcha de la propuesta se fundamenta en los siguientes aspectos:

- **Factibilidad Institucional:** Debe existir la intención de la gerencia en adoptar el modelo de madurez para la selección de una metodología de desarrollo de software que permita agilizar los procesos para el manejo de proyectos.
- **Factibilidad Técnica:** Permitirá saber si el modelo propuesto, por el cual se llevó a cabo este proyecto de investigación, puede ser utilizado por el

personal que labora en las Oficinas de Tecnología y Sistemas de la Administración Pública Nacional.

- **Factibilidad Operativa:** Brindarle al recurso humano una herramienta de fácil comprensión y utilización que permita definir desde el inicio los pasos a seguir para llevar a cabo un proyecto de software.

Por último, los aspectos relevantes de la propuesta del modelo de madurez para la selección de una metodología de desarrollo de software están orientados hacia una concientización en la mejora del proceso de escogencia de las metodologías que mejor se adecúen a los proyectos.

5.7 Definición de términos importantes

Según PaulK (1994) y como punto de partida se definen algunos términos manejados por el Instituto de Ingeniería de software (SEI):

5.7.1 Proceso de software

Con junto de actividades, métodos, prácticas y transformaciones para desarrollar y mantener software y productos asociados.

5.7.2 Capacidad de un proceso

El rango de resultados esperados que se pueden obtener tras seguir un proceso.

5.7.3 Madurez de proceso de software

Es el punto hasta el cual un determinado proceso es explícitamente definido, administrado, medido, controlado y efectivo.

La madurez es el potencial de crecimiento de la capacidad del proceso, además de indicar la certeza de la selección de una metodología de desarrollo de software como la consistencia con la cual es aplicada en los proyectos a lo largo de la misma.

5.7.4 Nivel de madurez

Plataforma bien definida desde la cual podremos obtener un proceso maduro de software.

A medida que una organización adquiere madurez en los procesos de selección de selección de metodologías de desarrollo de software, ésta lo institucionaliza a través de estructuras organizacionales, políticas y estándares.

Los niveles definen una escala para medir la madurez y evaluar la capacidad de los procesos de software.

5.7.5 Áreas clave de proceso

Grupo de actividades relacionadas que cuando se llevan a cabo en conjunto se alcanzan metas, las cuales son consideradas importantes para aumentar la capacidad del proceso.

5.8 Estructura

El modelo de madurez cuenta con cinco niveles de madurez, los cuales son progresivos y no autónomos y están organizados conforme su importancia o prioridad.

Estos niveles indican la capacidad del proceso. Los niveles están contienen áreas claves de proceso las cuales permiten alcanzar ciertas metas consideradas importantes para la mejora del proceso.

5.8.1 Niveles de madurez

La estructura del modelo está basada en los cinco niveles de madurez, donde cada uno de estos niveles indica que tan capaz en un proceso.

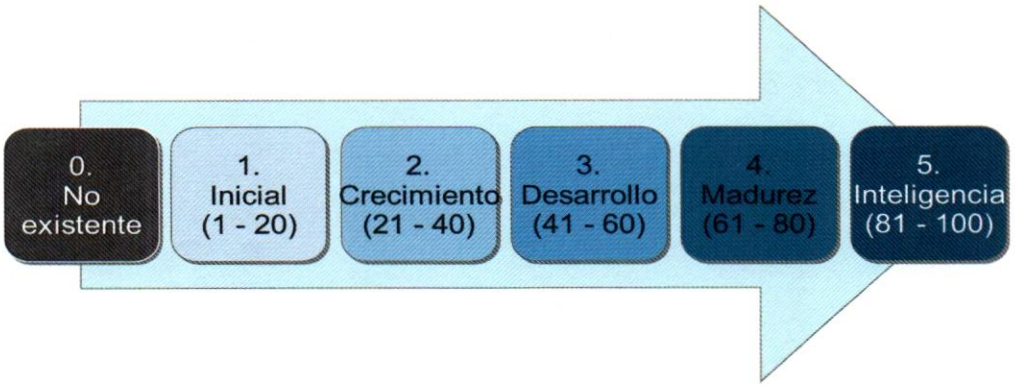


Figura 12. Niveles de madurez

5.8.2 Descripción de los niveles del modelo de madurez

Nivel 0, No existente

- No se usan técnicas de selección de metodologías de desarrollo de software y la organización no toma en cuenta los impactos al negocio asociados con las fallas de desarrollo en el proyecto.
- No se toman en cuenta los impactos en el negocio asociados a las vulnerabilidades de seguridad y a las incertidumbres del desarrollo de proyectos.
- No existe documentación de los sistemas.

Nivel 1, Inicial

- No se alcanzan las metas definidas ni en tiempo, ni costo, ni recursos planeados.
- Se alcanza el propósito del proceso de manera inconsistente.

Nivel 2, Crecimiento

- La organización está en proceso de desarrollar y utilizar algunas técnicas y métodos proyecto por proyecto.
- Los proyectos de TI han definido objetivos técnicos y de negocio de manera informal.

- Existe un proceso de administración de cambio informal y la mayoría de los cambios siguen este enfoque; sin embargo, el proceso no está estructurado, es rudimentario y propenso a errores.
- La necesidad de contar con una organización estructurada y una administración de proveedores se comunica, pero las decisiones todavía dependen del conocimiento y habilidades de individuos clave.
- Los niveles de servicio están acordados pero son informales y no están revisados.
- Los sistemas de información contemplan la mayor parte de las actividades institucionales, y en el desarrollo de algunos de ellos se han incorporado componentes digitales, tomando como referencia la necesidad de una gestión documental que permita satisfacer los requerimientos de la institución.
- Se han instaurado algunos procesos para la generación de información que responda a las necesidades de los diferentes usuarios.

Nivel 3, Desarrollo

- El proceso y la metodología de desarrollo de software han sido establecidos y comunicados.
- Los proyectos de TI se monitorean, con puntos clave, cronogramas y mediciones de presupuesto y desempeño definidos y actualizados.
- Se dan soluciones temporales a los problemas y los procesos a menudo se omiten o se hacen a un lado.

Nivel 4, Madurez

- La gerencia requiere que se revisen métricas y lecciones aprendidas estandarizadas y formales después de terminar cada proyecto.
- Las mejoras al proceso de selección de metodologías de desarrollo de software se formalizan y comunican y los miembros del equipo reciben entrenamiento sobre estas mejoras.

- La gerencia de TI implementa una estructura organizacional de proyectos con roles, responsabilidades y criterios de desempeño documentados.
- El proceso de administración de cambio se desarrolla bien y es consistente para todos los cambios, y la gerencia confía que hay excepciones mínimas.

Nivel 5, Inteligencia

- Se encuentra implantada una metodología comprobada de ciclo de vida de proyectos, la cual se refuerza y se integra en la cultura de la organización completa.
- Se ha implantado una iniciativa continua para identificar e institucionalizar las mejores prácticas de selección de una metodología de desarrollo de software.
- Se ha definido e implantado una estrategia de TI para contratar el desarrollo y los proyectos operativos.
- El proceso de administración de cambios se revisa con regularidad y se actualiza para permanecer en línea con las buenas prácticas.

5.9 Áreas de proceso

Para desarrollar este modelo se reunieron las mejores prácticas y estándares establecidos en la Ingeniería de software.

Se definen un conjunto de áreas de proceso como claves para lograr el mejoramiento en la selección de la metodología de desarrollo de software que mejor se adapte al proyecto a ejecutar.

Un nivel de madurez será obtenido si y sólo si todas las áreas de proceso comprendidas en él han sido cubiertas satisfactoriamente. Para saber si un área ha sido cubierta basta con que las metas especificadas se hayan alcanzado.

El único nivel de madurez que no tiene áreas de proceso es el nivel No existente. Entonces, a partir del nivel 1, el Inicial, que se definen las áreas por nivel. En total son 14 áreas que conforman el modelo de madurez y están agrupadas en cinco

categorías: gerencia de TI, impacto financiero, riesgos tecnológicos, sistemas de información y metodologías de desarrollo de software

5.10 Instrumento de evaluación

En los capítulos previos se presentaron los aspectos relacionados los niveles de madurez y se detalló toda la teoría relacionada. Sin embargo, esto por sí solo no constituye la solución al problema planteado en esta investigación, para ello es necesario desarrollar una guía de diagnóstico basado en las características de los niveles de madurez, el cual permite realizar la calificación de la situación de la Oficina de Gestión de Tecnología y Sistemas de la Administración Pública Nacional para poder aplicar sus principios de mejora.

Al terminar el diagnóstico se tiene la capacidad de situar a la oficina dentro de uno de los cinco niveles de capacidad del modelo.

5.10.1 Escalamiento tipo Likert

Según Hernández (2010) citado por Hernández, F. (2011), este método fue desarrollado por Rensis Likert en 1952; sin embargo, aún se encuentra vigente. Consiste en un conjunto de ítems presentados en forma de afirmaciones o juicios, ante los cuales se pide o se mide la reacción de los participantes. Es decir, se presenta cada afirmación y se solicita al sujeto que extreme su reacción eligiendo uno de cinco puntos o categorías de la escala. A cada punto se le asigna un valor numérico. Así, el participante obtiene una puntuación respecto a la afirmación y al final su puntuación total, sumando las puntuaciones en relación con todas las afirmaciones.

Según Hernández (2010) citado por Hernández, F (2011), las puntuaciones se obtienen sumando los valores alcanzados respecto a cada frase. Por ello se denomina escala aditiva. Una puntuación se considera alta o baja según el número de afirmaciones. En la escala de Likert a veces se califica el promedio resultante en la escala mediante la fórmula PT/NT (donde PT es la puntuación total de la escala y NT el número de afirmaciones), donde la puntuación se analiza en el continuo 1-5.

En este trabajo se utilizó la escala de Likert donde el promedio resultante se basa en la siguiente escala:

Tabla 6. Escala de Likert

Inicial	Crecimiento	Desarrollo	Madurez	Inteligencia
1	2	3	4	5

Fuente: Adaptado de Hernández, F. (2011)

5.10.2 Codificación de las respuestas

Según Hernández (2010) citado por Hernández, F (2011), una vez recolectados los datos, éstos deben codificarse. Para ello es necesario transformar las respuestas en símbolos o en valores numéricos. Los datos deben ser preparados para su análisis, porque de lo contrario solo se contaría con número de respuestas.

A cada prioridad se le asigna un valor como se indica en el siguiente cuadro:

Tabla 7. Valorización de prioridades

Prioridad	Valor
1	20
2	40
3	60
4	80
5	100

Fuente: Adaptado de Hernández, F. (2011)

Aplicando la fórmula PT/NT:

PT= Suma de los valores obtenidos (prioridad * valor).

NT= 100 el valor máximo de afirmaciones.

5.10.3 Valoración de las preguntas

Cuando se realiza la evaluación a cada ítem se le asigna un valor que corresponde a la respuesta obtenida.

Tabla 8. Valorización de las preguntas

Nro. Preg.	Valor	Si	No	Escala
1	4			4 puntos si es menor a 6 meses
2	2	2	0	
3	2	2	0	
4	2	2	0	
5	4			4 puntos si es menor a 2
6	2	2	0	
7	4			4 puntos si es menor a 2
8	2	2	0	
9	2	2	0	
10	2	2	0	
11	2	2	0	
12	2	2	0	
13	4			4 puntos si es Específico
14	4			4 puntos si es bajo
15	2	2	0	
16	4			4 puntos
17	2	2	0	
18	4			4 puntos si es Alto contacto
19	2	2	0	
20	4			4 puntos si es igual a 5
21	4			4 puntos si es altamente centralizados
22	4			4 puntos si tiene procesos de TI
23	4			4 puntos si es centralizados
24	4			4 puntos si es mayor a 4
25	4			4 puntos si es mayor a 4
26	4			4 puntos si es mayor a 4
27	4			4 puntos si es igual a 5
28	4			4 puntos si es igual a 5
29	4			4 puntos si es igual a 5
30	4			4 puntos si es mayor a 3

5.10.4 Parámetros de valoración de los niveles de madurez

A continuación se detallan los parámetros de valoración del modelo de madurez y sus respectivas preguntas.

Tabla 9. Parámetros de valorización

Niveles de madurez	% Valoración
Nivel 0: No existente	0
Nivel 1: Inicial /Ad Hoc	1 – 20
Nivel 2: Crecimiento	21 – 40
Nivel 3: Desarrollo	41 – 60
Nivel 4: Madurez	61 – 80
Nivel 5: Inteligencia	81 – 100

CAPÍTULO VI

6. VALIDACIÓN DEL MODELO DE MADUREZ PROPUESTO

El modelo de madurez propuesto fue validado a través de su aplicación en un caso de estudio en la Oficina de Gestión y Sistemas del Ministerio del Poder Popular para Industrias.

6.1 Aplicación

Se realizó la aplicación del modelo de madurez para seleccionar una metodología de desarrollo de software en la oficina mencionada anteriormente. Se analizan en esta sección los resultados de dicha aplicación.

6.1.1 Niveles de madurez por área de proceso

En la tabla 10 se aprecian los resultados obtenidos luego de aplicar la fórmula del promedio resultante de la escala de Likert.

Tabla 10. Niveles de madurez

	Prioridad 1		Prioridad 2		Prioridad 3		Prioridad 4		Prioridad 5		
ÍNDICE GENERAL DE MADUREZ	**Cant.**	**Valor**	**Cant.**	**Valor**	**Cant.**	**Valor**	**Cant.**	**Valor**	**Cant.**	**Valor**	**Nivel de madurez**
Sección 1 — GERENCIA DE TI											
1.1. Complejidad de la estructura	4	80	5	200	3	180	2	160	0	0	Crecimiento
1.2. Madurez de los procesos	1	20	8	320	4	240	1	80	0	0	Crecimiento
1.3. Disponibilidad de servicios TI	10	200	2	80	2	120	0	0	0	0	Inicial
1.4. Niveles de acuerdo de servicio	13	260	1	40	0	0	0	0	0	0	Inicial
Sección 2 — IMPACTO FINANCIERO											
2.1. Costos de desarrollo o implementación	8	160	3	120	2	120	1	80	0	0	Inicial

Sección 3 — RIESGOS TECNOLÓGICOS											
3.1. Combinación entre riesgo tecnológico y personal idóneo	7	7	4	160	1	60	1	80	1	100	Crecimiento
3.1. Problemas con los clientes internos	7	140	7	280	0	0	0	0	0	0	Crecimiento

Sección 4 — SISTEMAS DE INFORMACIÓN											
4.1. Alcance de los sistemas de información	7	140	4	160	2	120	1	80	0	0	Crecimiento
4.2. Calidad de la información	3	60	7	280	3	180	1	80	0	0	Crecimiento
4.3. Seguimiento de las tareas	4	80	8	320	2	120	0	0	0	0	Crecimiento
4.4. Estudios costo beneficio	13	260	1	40	0	0	0	0	0	0	Inicial

Sección 5 — METODOLOGIAS DE DESARROLLO DE SOFTWARE											
5.1. Concordancia entre metodología de desarrollo y proyectos	5	100	9	360	0	0	0	0	0	0	Crecimiento
5.2. Ajuste de los objetivos	3	60	4	160	5	300	2	160	0	0	Desarrollo
5.3. Planificación de actividades a desarrollar	2	40	2	80	10	600	0	0	0	0	Desarrollo
5.4. Metodología de desarrollo utilizada	14	280	0	0	0	0	0	0	0	0	Inicial

De estos resultados se puede inferir que la mayoría de las categorías se encuentran en la prioridad 1 y 2, que representa el primer y segundo peldaño en la escala de cinco niveles de mejoramiento de los procesos de software de una empresa, lo cual es importante para alcanzar competitividad organizacional.

6.1.2 Diagnóstico del grado de madurez

Una vez realizado el diagnóstico en la Oficina de Tecnología y Sistemas del Ministerio del Poder Popular para Industrias, el puntaje obtenido fue de 40. Aplicando los parámetros de valorización, se puede decir que se encuentra en el nivel de madurez, Crecimiento. Ver tabla 9).

Las principales características de este nivel son:

- La organización está en proceso de desarrollar y utilizar algunas técnicas y métodos proyecto por proyecto.

- La necesidad de contar con una organización estructurada y varias metodologías de desarrollo de software se comunica, pero las decisiones todavía dependen del conocimiento y habilidades de individuos clave.
- Los niveles de servicio están acordados pero son informales y no están revisados.

6.1.3 Diagnóstico por área

El diagnóstico por área se realizó, desglosando por áreas de preguntas. (Ver tabla 11)

Tabla 11. Parámetros de valorización

Área	Preguntas	Valores obtenidos
Estimación tiempos de desarrollo	1-4	2
Cantidad entregables	5-6	2
Primeros resultados	7-9	0
Introducir cambios	10-12	2
Definición de requerimientos	13	4
Riesgo tecnológico	14-15	6
Paradigma de desarrollo	16-17	4
Disponibilidad del cliente	18-19	0
Impacto financiero	20	4
Complejidad de la Gerencia de TI	21-25	10
Resistencia a la innovación de TI	26	0
Apoyo de la alta gerencia	27	0
Gerencia de proyectos	28	4
Compatibilidad con la metodología de desarrollo	29-30	2
	Total:	**40**

Y donde se obtuvo el siguiente gráfico. (Ver figura 13)

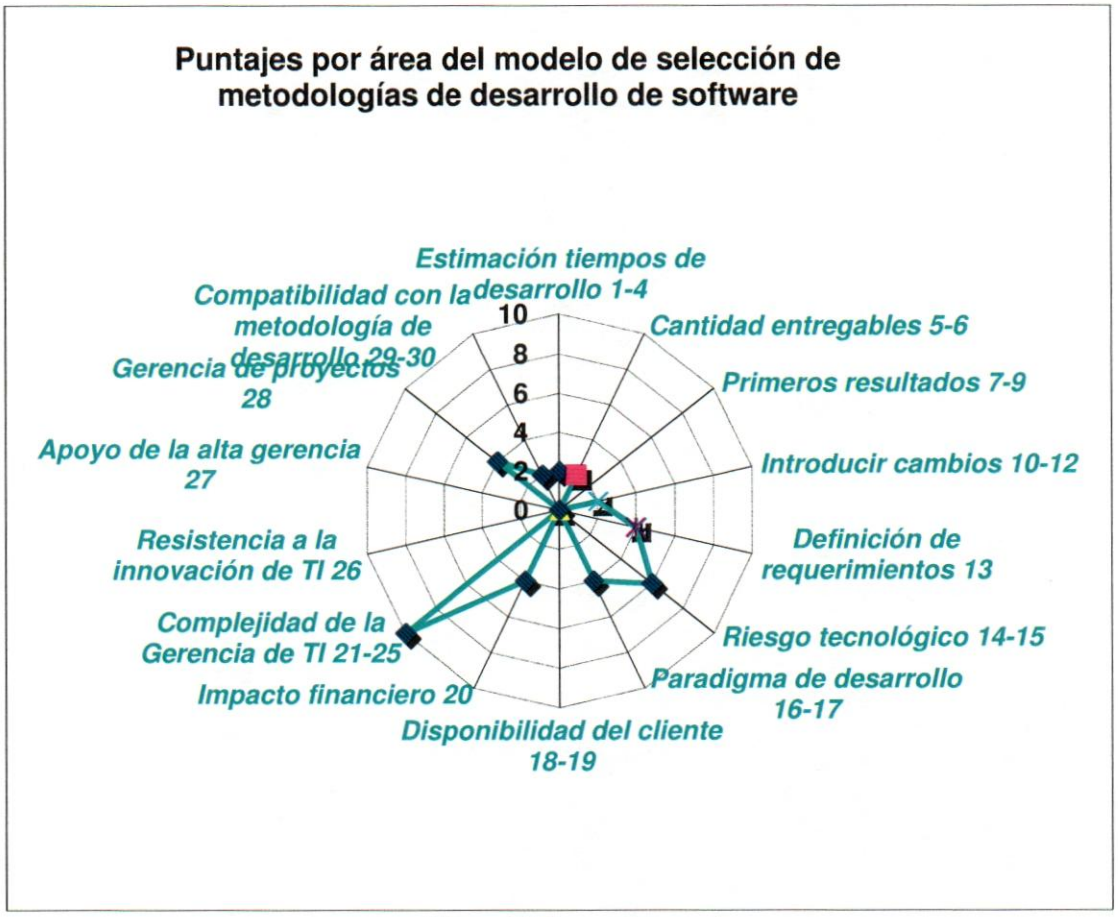


Figura 13.Respuestas por área

CAPÍTULO VII

7. CONCLUSIONES Y RECOMENDACIONES

7.1 Conclusiones

Para este proyecto, se realizó una propuesta que permite demostrar la necesidad de aplicar las mejores prácticas para llevar a la organización a un camino ordenado de mejoramiento al seleccionar metodologías de desarrollo de software. El objeto de este estudio fue lograr definir un modelo de madurez que permitiera iniciar un proceso de identificación de acciones a seguir con el objetivo fundamental de aumentar su rendimiento, mejorar sus procesos para fortalecer el negocio, enfrentar los retos de un entorno dinámico y más exigente.

También, representó un avance significativo esto debido a que el modelo de madurez constituyó un instrumento válido para realizar el diagnóstico inicial y el establecimiento de los criterios de mejora para seleccionar una metodología de desarrollo de software.

La problemática existente en las Oficinas de Tecnología y Sistemas de la Administración Pública Nacional resultó en una débil comprensión de la importancia de seleccionar una metodología de desarrollo de software que se adapte al proyecto a desarrollar. Existen carencias de conocimientos de las distintas metodologías existentes ya que solo utilizan una para todos los tipos de proyecto.

En cuanto a las deficiencias que originaron la problemática existente, se encontraron las siguientes: no se realizan inspecciones de código o revisiones técnicas formales, no se realiza un proceso formal de pruebas, escaso o deficiente control del progreso del proceso de desarrollo de software, excesiva e irracional presión de los plazos, poca gestión de los requisitos, escasa o tardía validación con el cliente. Por lo tanto, deben considerarse los conocimientos y técnicas especializadas, los recursos físicos, la formación y satisfacción de los empleados, así como las tecnologías de información.

Las etapas o niveles del modelo de madurez sobre la base de los estándares y mejores prácticas existentes, se logró definir con la ayuda de los expertos de TI en la Administración Pública Nacional y con base en Cobit 4.1. El modelo que se obtuvo consta de seis (6) niveles, los cuales son: (0) No existente, (1) Inicial, (2) Crecimiento, (3) Desarrollo, (4) Madurez y (5) Inteligencia. El modelo está acompañado de una guía de diagnóstico.

El modelo de madurez obtenido se validó mediante su aplicación en un caso de estudio en el Ministerio del Popular para Industrias, la verificación del modelo incrementó sustancialmente la probabilidad de que se ajustara al uso que se tenía previsto.

7.2 Recomendaciones

La utilización del modelo depende en gran medida de los esfuerzos promocionales que se hagan para estimular el uso del mismo, requiriendo para ello tiempo y esfuerzo de los Directores, Coordinadores de desarrollo y líderes de proyecto dentro de la organización. A continuación se detallan algunas actividades que se recomiendan seguirse para el éxito en la aplicación del modelo:

- Aplicar del modelo en otro ente de la Administración Pública Nacional, con la finalidad de garantizar la eficacia del mismo.
- Aplicar el modelo con cierta periodicidad como parte de un proceso de diagnóstico de la situación. Es importante destacar, que la madurez es un ciclo evolutivo que no tiene fin, por lo que es necesaria su aplicación, esto con la finalidad de medir avances y competitividad en materia de metodologías y desarrollos de sistemas de información.
- Por último, como trabajo a futuro se recomienda desarrollar un sistema automatizado que permita emular la función del modelo, para obtener los resultados de una forma más eficiente.

REFERENCIAS BIBLIOGRÁFICAS

Impresas

Arias, Fidias G. (1997). "**El proyecto de investigación. Guía para su elaboración**". (2da. Ed.) Caracas: Episteme

Balestrini, M. (2001). "**Cómo se elabora el proyecto de investigación**" (5ta. Ed.) Venezuela: OBL Servicio Editorial.

Avison, D y Fitzgerald, G. (1995). "**Information systems development: methodologies, techniques & tools**". (2da. Ed.). McGraw-Hill

Forouzan, Behrouz A. (2003) "**Introducción a la ciencia de la computación. De la manipulación de datos a la teoría de la computación**". International Thomson Editores. México, D.F.

Hernández, R., Fernández, C. y Pilar, B. (1998) "**Metodología de la investigación**" (2da. Ed.) México: McGraw-Hill.

Hurtado, J. (2010). "**El Proyecto de Investigación, Comprensión holística de la metodología y la investigación**". (6a Ed.).Caracas. Ediciones Quirón.

Tamayo y Tamayo, M. (1994). "**El proceso de la investigación científica**". (3ra. ed) México: Limusa

Web

Armadillo Integración Tecnológica. (2013). **MeRinde**. Recuperado el 13 de noviembre de 2013 de: http://merinde.net/index.php?option=com_content&task=view&id=38&Itemid=156

Advanced Quality Solutions. "**Cómo gestionar proyectos IT con éxito**". 2002. Recuperado abril 1, 2013 de: http://www.aqs.es/files/gestion_proyectos_con_exito.pdf

Amaro, S. y Valverde, J. (2007). "**Metodologías ágiles**". Trujillo, Perú. Recuperado abril 1, 2013 de: https://uvirtual.unet.edu.ve/pluginfile.php/268695/mod_resource/content/1/Metodologias%20Agiles.pdf

Canós, J., Letelier, P. y Penadés, M. (s.f). "**Metodologías ágiles de desarrollo de software**". Universidad Politécnica de Valencia. Valencia, España. Recuperado el 19 de noviembre de 2013 de: http://www.willydev.net/descargas/prev/TodoAgil.Pdf

CEISoft (CPTM), GIDYC (ULA), LISI(USB), FONACIT (2008). **"Estudio de la Industria Venezolana del Software. Informe técnico 2007" (Proyecto de Investigación Nro. 2005000165).** Recuperado el 13 de diciembre de 2013 de: http://ldc.usb.ve/~abianc/materias/ci4712/METODIUS-Estudio-necesidades-INS-2007.pdf

Citón, M. (2006). "**Método ágil Scrum. Aplicado al desarrollo de un software de trazabilidad**". Universidad de Mendoza. Argentina. Recuperado abril 26, 2013 de: http://www.um.edu.ar/catedras/claroline/backends/download.php?url=L01ldG9kb3NfQWdpbGVzL01ldG9kb19BZ2lsX1NjcnVtLnBkZg%3D%3D&cidReset=true&cidReq=II0162004

Cobit 4.1 (2007) IT Governance Institute. Impreso en Estados Unidos de América. Recuperado abril 1, 2013 de: http://www.itgi.org/Content/NavigationMenu/Members_and_Leaders/COBIT6/Obtain_COBIT/CobiT4_Espanol.pdf

Fernández, V. (2006). "**Desarrollo de sistemas de información. Una metodología basada en el modelado**". Ediciones Universidad Politécnica de Cataluña. Barcelona, España. Recuperado abril 24, 2013 de: http://books.google.co.ve/books?hl=en&lr=&id=pTTQ735ac1EC&oi=fnd&pg=PA7&dq=metodología+de+desarrollo+de+sistemas&ots=zehfX1Kac5&sig=qQ_X3GlJUIn-TJNsxHv10jSPvOg#v=snippet&q=sistemas&f=false

Fleittman J. (s/f). "**La importancia de los sistemas de informacion y control en la empresa**". Recuperado abril 24, 2013 de: http://www.fleitman.net/articulos/sisInfo.pdf

Forouzan, Behrouz A. 2003 "**Introducción a la ciencia de la computación. De la manipulación de datos a la teoría de la computación**". International Thomson Editores. México, D.F. Recuperado el 17 de noviembre de 2013 de: http://books.google.co.ve/books?id=xu4jcMBih8QC&pg=PT5&lpg=PT5&dq=Forouzan,+Behrouz+A.+2003+%E2%80%9CIntroducci%C3%B3n+a+la+ciencia+de+la+computaci%C3%B3n.+De+la+manipulaci%C3%B3n+de+datos+a+la+teor%C3%ADa+de+la+computaci%C3%B3n&source=bl&ots=Yq8cpAFqKs&sig=vhiaZPXZ7a0nlCCEtZ-jiCYLtR0&hl=en&sa=X&ei=PH-eUr6ZFsnskQfJ4IFg&ved=0CEYQ6AEwAw#v=onepage&q=Forouzan%2C%20Behrouz%20A.%202003%20%E2%80%9CIntroducci%C3%B3n%20a%20la%20ciencia%20de%20la%20computaci%C3%B3n.%20De%20la%20manipulaci%C3%B3n%20de%20datos%20a%20la%20teor%C3%ADa%20de%20la%20computaci%C3%B3n&f=false

Gabardini, J y Campos, L. (2004). "**Balanceo de metodologías orientadas al plan y ágiles. Herramientas para la selección y adaptación**". PMI Global Congress Proceedings. Buenos Aires, Argentina. Recuperado abril 1, 2013 de: http://www.rmya.com.ar/Download/Paper_BMOPA.pdf

Kendall, K y Kendall, J. (2005). "Análisis y diseño de sistemas". (6ta. Ed). Mexico. Editorial Prentice Hall. Recuperado abril 24, 2013, de: http://books.google.co.ve/books?id=5-rZA0FggusC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Hernández, Alejandro. (s/f). "**Los sistemas de información: Evolución y desarrollo**". Departamento de Economía y Dirección de empresas. Universidad de Zaragoza. Recuperado abril 24, 2013, de: http://dialnet.unirioja.es/servlet/articulo?codigo=793097

Méndez, E. (2006). "**Modelo de evaluación de metodologías para el desarrollo de software**". Trabajo de grado. Universidad Católica Andrés Bello, Caracas, Venezuela. Recuperado abril 1, 2013, de: http://biblioteca2.ucab.edu.ve/anexos/biblioteca/marc/texto/AAQ7365.pdf

PaulK, M. 1994. "A comparison of ISO 9001and the Capability Maturity Model for Software". Technical report CMU/SEI-94-TR-012

PMI (2008). "**Guía de los fundamentos para la Dirección de Proyectos (Guía del PMBOK)**". (4ta. Edición). Project Management Institute. Recuperado el 01 de julio de 2013 de: http://www.tfinnova.es/userfiles/file/pmbok%204%20edicion_Espa%C3%B1ol.pdf

Pressman, R. 2002. "Ingeniería del software. Un enfoque práctico". McGraw Hill: Quinta edición México. Recuperado el 17 de noviembre de 2013, de: http://www.slideshare.net/jdbg16/ingenieria-de-software-un-enfoque-prctico-pressman-5th-ed

Real Academia Española (2011). Metodología. En Diccionario de la lengua española (23ª. ed.). Recuperado el 27 de junio de 2013 de: http://lema.rae.es/drae/?val=metodologia

Rivero, D., Montilva, J., Granados, G.,Barrios, J., Besembel, I. y Sandia, B. (s/f). **"La industria nacional del software en Venezuela". Universidad de Los Andes, Mérida, Venezuela. Grupo de Investigación en Ingeniería de Datos y Conocimiento (GIDyC), Centro de Excelencia en Ingeniería de Software (CEISoft), Coordinación de Estudios Interactivos a Distancia (CEIDIS). Recuperado el 13 de diciembre de 2013 de:** http://www.saber.ula.ve/bitstream/123456789/15822/1/09_INS_EVETIS2007.pdf

Sommerville, I. (2006) "Ingeniería del software". (7ma. Ed). España. Editorial Pearson Educación. Recuperado abril 24, 2013, de: http://www.google.co.ve/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&sqi=2&ved=0CCsQFjAA&url=http%3A%2F%2Ftodocarrera.googlecode.com%2Ffiles%2FSommerville.pdf&ei=2tF6UZSVNIfq8wTJ9YHQAg&usg=AFQjCNGBtNzTqCBj-SlQioblwp9v1LHxKg&sig2=E1SXQJDfPrPxGRxRjrZ2vw&bvm=bv

Tinoco, O, Rosales, P. y Salas, J. "**Criterios de selección de metodologías de desarrollo de software**". Industrial Data, vol. 13, núm 2. julio, 2010, pp. 70-74. Recuperado abril 1, 2013 de: http://www.redalyc.org/articulo.oa?id=81619984009 Universidad Nacional Mayor de San Marcos, Perú.

Trabajos de grado

Angles, R. y Matos, G. (2004). **Modelo de apoyo en la selección de la metodología para un Desarrollo de Software, basado en la gerencia del conocimiento y soportado en tecnología de información.** Universidad Católica Andrés Bello, Caracas.

Bastardo, M. (2006). **Selección de una metodología para la Gerencia de Proyectos de Desarrollo de Software.** Universidad Católica Andrés Bello, Caracas.

Bonillo, P. (2003). **Modelo sistémico de evaluación financiera de proyectos de inversión de tecnología de la información en Venezuela**. Universidad Yacambú. Barquisimeto, Edo. Lara, Venezuela. Recuperado el 01 de julio de 2013 de: http://www.oocities.org/es/pedrobonillotesis/proyecto/index.html

Hernández, F. (2011). **Desarrollo de un modelo de madurez para las contingencias y recuperación de la información en las pequeñas y medianas empresas del sector salud en Venezuela**. Universidad Católica Andrés Bello, Caracas.

Méndez, E. (2006). **Modelo de evaluación de metodologías para el desarrollo de software**. Universidad Católica Andrés Bello, Caracas.

FUENTES LEGALES

Constitución de la República Bolivariana de Venezuela (1999). Gaceta Oficial N°. 36.860 del 30 de diciembre de 1999. Recuperado el 01 de julio del 2013 de: http://www.tsj.gov.ve/legislacion/constitucion1999.htm

Ley de Interoperabilidad (2012). Gaceta Oficial N° 39.945 del 15 de junio del 2012. Recuperado el 01 de julio del 2013 de: http://interoperabilidad.gobiernoenlinea.gob.ve/index.php/conceptos/sobre-promociones/promociones/96-proyecto-de-interoperabilidad

ANEXO A

CONSTITUCIÓN DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA

(Publicada en Gaceta Oficial N°. 36.860 del 30 de diciembre de 1.999)

Artículo 110

El Estado reconocerá el interés público de la ciencia, la tecnología, el conocimiento, la innovación y sus aplicaciones y los servicios de información necesarios por ser instrumentos fundamentales para el desarrollo económico, social y político del país, así como para la seguridad y soberanía nacional. Para el fomento y desarrollo de esas actividades, el Estado destinará recursos suficientes y creará el sistema nacional de ciencia y tecnología de acuerdo con la ley. El sector privado deberá aportar recursos para las mismas. El Estado garantizará el cumplimiento de los principios éticos y legales que deben regir las actividades de investigación científica, humanística y tecnológica. La ley determinará los modos y medios para dar cumplimiento a esta garantía.

LEY INTEROPERABILIDAD

(Gaceta Oficial N° 39.945 del 15 de junio de 2012. Decreto N°. 9.051 15 de junio de 2012.)

Artículo 1. Objeto

El presente Decreto con Rango, Valor y Fuerza de Ley tiene por objeto establecer las bases y principios que regirá el acceso e intercambio electrónico de datos, información y documentos entre los órganos y entes del Estado, con el fin de garantizar la implementación de un estándar de interoperabilidad.

Artículo 2. Ámbito de Aplicación

Están sometidos a la aplicación de las disposiciones del presente Decreto con Rango, Valor y Fuerza de Ley:

1. Los órganos del Poder Público Nacional, Estadal y Municipal.

2. Los institutos públicos nacionales, estadales, distritales y municipales.

3. El Banco Central de Venezuela.

4. Las Universidades públicas nacionales autónomas y experimentales, así como cualquier otra institución del sector universitario de naturaleza pública.

5. Las demás personas de derecho público nacionales, estadales, distritales y municipales.

6. Las sociedades de cualquier naturaleza en las cuales las personas a que se refieren los numerales anteriores tengan una participación en su capital social superior al cincuenta por ciento (50%), las que se constituyan con la participación de aquéllas, o que a través de otro mecanismo jurídico, tenga el control de sus decisiones.

7. Las fundaciones y asociaciones civiles y demás instituciones creadas con fondos públicos, o que sean dirigidas por las personas a que se refieren los numerales anteriores, o en las cuales tales personas designen sus autoridades, o cuando los aportes presupuestarios o contribuciones efectuados en un ejercicio, por una o varias de las personas a que se refieren los numerales anteriores, representen el cincuenta por ciento (50%) o más de su presupuesto.

8. Los demás entes de carácter público.

Artículo 3. Fines

El presente Decreto con Rango, Valor y Fuerza de Ley tiene los siguientes fines:

1. Establecer un estándar de interoperabilidad entre los órganos y entes del Estado.

2. Establecer las condiciones necesarias para el desarrollo y adopción de planes y proyectos que garanticen el acceso e intercambio electrónico de datos, información y documentos entre los órganos y entes del Estado.

3. Promover el desarrollo de sistemas de información interoperables adecuados para los procesos del Estado y la satisfacción de las necesidades de los ciudadanos.

4. Promover el desarrollo de una Plataforma Nacional de Servicios de Información Interoperables que provea un acceso uniforme de datos, información y documentos entre los órganos y entes del Estado.

5. Promover el desarrollo de un modelo nacional para el intercambio, publicación e interpretación de los datos, información y documentos, que apoye el establecimiento de políticas, lineamientos y estrategias públicas.

6. Garantizar un adecuado nivel de interoperabilidad en los sistemas de información utilizados por los órganos y entes del Estado.

7. Coadyuvar en la gobernabilidad del Estado con el fortalecimiento, seguimiento y evaluación de las políticas, planes, programas y proyectos enmarcados en los objetivos estratégicos de la nación.

8. Contribuir con la mejora del funcionamiento interno de los órganos y entes del Estado, impulsando una mayor eficiencia y eficacia en las actividades que soportan los servicios que éstos prestan.

9. Coadyuvar en la ordenación, coordinación, cooperación, armonización y racionalización de la acción pública de los órganos y entes del Estado.

10. Coadyuvar en la simplificación de los trámites que realizan los ciudadanos ante los órganos y entes del Estado.

Capítulo I.

De La Interoperabilidad

Artículo 22. Acceso e Intercambio de Datos, Información y Documentos

Los órganos y entes del Estado están obligados a permitir entre sí, el acceso, intercambio y reutilización, por medios electrónicos, de los datos de autoría, información y documentos de acceso público que posean, en los términos y condiciones establecidas en el presente Decreto con Rango, Valor y Fuerza de Ley y demás normativa aplicable.

Artículo 24. Elementos de los Sistemas Informáticos

Los órganos y entes del Estado deberán incorporar en sus sistemas informáticos todos los elementos técnicos requeridos por el operador de la interoperabilidad para realizar el proceso de intercambio electrónico de datos, información y documentos.

SOFTWARE ENGINEERING INSTITUTE (SEI)

Es un instituto federal estadounidense de investigación y desarrollo, fundado por Congreso de los Estados Unidos en 1984 para desarrollar modelos de evaluación y mejora en el desarrollo de software, que dieran respuesta a los problemas que generaba al ejército estadounidense la programación e integración de los sub-sistemas de software en la construcción de complejos sistemas militares. Financiado por el Departamento de Defensa de los Estados Unidos y administrado por la Universidad Carnegie Mellon.

Es un referente en Ingeniería de Software por realizar el desarrollo del modelo SW-CMM (1991) que ha sido el punto de arranque de todos los que han ido formando parte del modelo que ha desarrollado sobre el concepto de capacidad y madurez, hasta el actual CMMI.

PROJECT MANAGEMENT INSTITUTE (PMI)

El Project Management Institute (PMI) considera la norma como una referencia fundamental en el ámbito de la dirección de proyectos para sus certificaciones y programas de desarrollo profesional.

En su carácter de referencia fundamental, la norma no está completa ni abarca todos los conocimientos.

Se trata de una guía, más que de una metodología. Se pueden usar diferentes metodologías y herramientas para implementar el marco de referencia.

Además de las normas que establecen pautas para los procesos, herramientas y técnicas de la dirección de proyectos, el Code of Ethics and Professional Conduct del Project Management Institute sirve de guía a los profesionales de la dirección de proyectos y describe las expectativas que tienen de sí mismos y de los demás.

El Code of Ethics and Professional Conduct del Project Management Institute precisa las obligaciones básicas de responsabilidad, respeto, imparcialidad y honestidad. Requiere que quienes se desempeñan en este ámbito demuestren compromiso con la conducta ética y profesional. Conlleva la obligación de cumplir con leyes, regulaciones y políticas profesionales, y de la organización. Puesto que los

profesionales provienen de culturas y orígenes diversos, el Code of Ethics and Professional Conduct se aplica a nivel mundial. En el trato con los interesados, los profesionales deben comprometerse a realizar prácticas justas y honestas, y a mantener relaciones respetuosas. El Code of Ethics and Professional Conduct del Project Management Institute está publicado en el sitio Web del PMI (http://www.pmi.org). La aceptación del código es requisito para la certificación PMP® del PMI. (PMI, 2008)

PROJECT MANAGEMENT BODY OF KNOWLEDGE (PMBOK)

La Guía del PMBOK® proporciona pautas para la dirección de proyectos tomados de forma individual. Define la dirección de proyectos y otros conceptos relacionados, y describe el ciclo de vida de la dirección de proyectos y los procesos conexos.

La Guía del PMBOK® también proporciona y promueve un vocabulario común en el ámbito de la profesión de la dirección de proyectos, para analizar, escribir y aplicar conceptos de la dirección de proyectos. Un vocabulario estándar es un elemento esencial en toda disciplina profesional. (PMI, 2008)

ANEXO B

CUESTIONARIO

Estimado participante:

El presente cuestionario tiene por objeto recabar información para el desarrollo de la investigación que se titula: "Modelo de madurez para la selección de una metodología de desarrollo de software".

Quisiera pedir su apoyo para que conteste algunas preguntas, que a lo sumo le tomará quince (15) minutos. Sus respuestas serán tratadas de forma confidencial y anónimo, por lo que puede sentirse libre de expresar su opinión e ideas. La información que Usted proporcione será expresada en la investigación en términos globales y no individualmente, los resultados serán compartidos con los participantes garantizando la confidencialidad de la información.

Las personas seleccionadas para responder este cuestionario, fueron seleccionadas por el tipo de actividad laboral dentro del ente gubernamental, debido a que se considera que pueden aportan información valiosa al estudio.

Se le agradece que responda cada pregunta con la mayor objetividad posible, ya que no existen respuestas correctas o incorrectas.

El cuestionario consta de preguntas tipo cerradas (Si o No) y otras, en donde debe indicar prioridad donde 1 es la máxima y 5 la menor, las cuales deben ser marcadas con una "X" y seleccionar solo una de ellas.

Su valiosa colaboración será agradecida, para el éxito del desarrollo de ésta investigación.

Apellidos y Nombres: _____

Institución donde se desempeña: _____

Cargo: _____

Estimación de los tiempos de desarrollo: Actividad de planificación del proyecto de software que intenta determinar cuánto tiempo será necesario para construir un sistema o producto

1. Ha estado Usted involucrado en desarrollos de software a:

   ☐ Largo plazo (Mayor a 6 meses)

   ☐ Mediano (Entre 3 y 6 meses)

   ☐ Corto (Menos de 3 meses)

2. ¿Existe una planificación de las actividades a desarrollar dentro del proyecto?

   ☐ Si ☐ No

3. ¿Se define el tiempo de entrega para cada una de las actividades planificadas?

   ☐ Si ☐ No

4. Una vez realizada las actividades de planificación y estimación, ¿Conoce Usted Cuál es la fecha estimada de finalización del proyecto o sistema a desarrollar?

   ☐ Si ☐ No

Cantidad requerida de entregables: Resultado del esfuerzo del trabajo ejecutado por el equipo del proyecto
Documentar es una tarea tan necesaria e importante como el desarrollo del software

5. Considera Usted que la documentación entregada es:

   ☐ Buena

   ☐ Regular

   ☐ Mala

6. ¿Se realizan seguimiento de las tareas, los plazos, los costos y expectativas, con el fin de controlar los desarrollos de software, de tal forma que se eviten los retrasos en los proyectos?

   ☐ Si ☐ No

Tiempo para obtener los primeros resultados: Actividad planificada para la obtención de los primeros resultados del desarrollo del software

7. Los plazos de entrega de los primeros resultados se planifican a:

   ☐ Largo plazo (Mayor a 6 meses)

   ☐ Mediano (Entre 3 y 6 meses)

   ☐ Corto (Menos de 3 meses)

8. ¿Los primeros resultados obtenidos del desarrollo de software, son medidos contra los planes?

   ☐ Si ☐ No

9. ¿Si existe discrepancia entre los resultados y los objetivos, se toman a tiempo las acciones correctivas?

   ☐ Si ☐ No

Introducir cambios o nuevos requerimientos durante el desarrollo: Realinear el software según las necesidades del cliente o evoluciones del mercado

10. ¿Se definen procesos de control de cambios o ajuste de los objetivos?

    ☐ Si ☐ No

11. ¿Cuando la gestión de los requisitos no se hace bien, aparecen altos niveles de re-trabajo a lo largo del proyecto?

    ☐ Si ☐ No

12. ¿Las actividades para el aseguramiento de calidad del software son detalladas en Métricas de software para el control del proyecto, verificación y validación del software a lo largo del ciclo de vida?

    ☐ Si ☐ No

Nivel de definición de requerimientos: Los requerimientos proporcionan una vía en todo desarrollo de software. Obtener requerimientos de calidad demuestra que el trabajo realizado culminará con éxito

13. La definición del alcance del proyecto es:

☐ Específico ☐ Ambiguo

Riesgo tecnológico: Mide cual es la exposición del emprendedor a los riesgos tecnológicos derivados de la necesidad de acometer fuertes inversiones para asegurar la viabilidad de su proyecto empresarial en un plazo determinado de tiempo o la necesidad de formar a sus empleados en el uso de la tecnología

14. La combinación entre riesgo tecnológico y personal idóneo que componen la estrategia en la organización es:

☐ Alto (Combinación tecnológica no probada)

☐ Medio (Poca experiencia en el uso de la tecnología)

☐ Bajo (Tecnología madura y ampliamente utilizada)

15. Se presentan problemas con los clientes internos:

☐ Si ☐ No

Paradigma de desarrollo (lenguaje de desarrollo a utilizar): Técnica que permite el desarrollo de un sistema o software de manera exitosa y que permite dar solución a un problema

16. La metodología utilizada y aplicada para el desarrollo de diversos software es:

☐ Estructurado ☐ Orientados a objeto

17. La metodología utilizada es fácil aprender o enseñar

☐ Si ☐ No

Disponibilidad del cliente: Es el tiempo que dedica el cliente para involucrarse en el sistema o software a desarrollar

18. Las acciones proporcionadas por los clientes para continuidad en las actividades en el desarrollo del software son:

☐ Alto contacto (Presencia y participación activa)

☐ Contacto moderado (Menor presencia y participación, solo para evaluación)

☐ Bajo contacto (Implica muy poca participación, solo al principio y al final del proceso)

19. Grado de satisfacción

☐ Si ☐ No

Impacto financiero

20. El valor de los costos de desarrollo o implementación de los proyectos son:

☐ Menor que el 25% del valor deseado

☐ Menor que el 50% del valor deseado

☐ Menor que el 75% del valor deseado

☐ Igual al valor deseado

☐ Mayor al valor deseado

Complejidad de la Gerencia de TI: Es la relación de la gestión de las organizaciones y su conocimiento que permite tener una visión más clara hacia donde deben orientarse los gerentes para que sus instituciones logren niveles adecuados de competitividad

21. ¿Qué tan compleja es la estructura organizacional (organigrama)?

☐ Altamente Descentralizados

☐ Descentralizados

☐ Dispersos

☐ Centralizados

☐ Altamente Centralizados

22. ¿Cuál es el estado de desarrollo de los procesos de negocios en la organización?

- ☐ Tiene procesos de TI
- ☐ Tiene y usa los procesos de TI
- ☐ Tiene, usa, documenta los procesos de TI
- ☐ Tiene, usa, documenta, administra los procesos de TI
- ☐ Tiene, usa, documenta, administra y optimiza los procesos de TI

23. Dispersión de los usuarios

- ☐ Altamente Descentralizados
- ☐ Descentralizados
- ☐ Dispersos
- ☐ Centralizados
- ☐ Altamente Centralizados

24. Los servicios de TI están disponibles y funcionan correctamente siempre que los clientes y usuarios deseen hacer uso de ellos, con una frecuencia de:

- ☐ 7x24 díasxhora
- ☐ 7x15 díasxhora
- ☐ 7x12 díasxhora
- ☐ 6x9 díasxhora
- ☐ 5x8 díasxhora

25. En los niveles de acuerdo de servicio (SLA), la provisión de los servicios acordados se establece en:

- ☐ Inmediato o menor a 30 minutos
- ☐ 4 horas
- ☐ Mismo día garantizado
- ☐ Próximo día garantizado
- ☐ Próximo día o mejor esfuerzo

Resistencia a la innovación de TI: Resistencia al cambio y la falta de visión integral acerca de los proyectos

26. Los planes para la renovación periódica de la infraestructura de TI están:

- ☐ Completamente definidas
- ☐ Casi todas definidas
- ☐ Medianamente definidas
- ☐ Poco definidas
- ☐ No están definidas

Apoyo de la alta gerencia

27. ¿Se realizan los requerimientos de estudios de costo-beneficio para definir la factibilidad de los proyectos a ser desarrollados?

- ☐ Siempre
- ☐ Casi siempre
- ☐ Algunas veces
- ☐ Pocas veces
- ☐ Nunca

Gerencia de proyectos: Es la disciplina de organizar y administrar los recursos, de forma tal que un proyecto dado sea terminado completamente dentro de las restricciones de alcance, tiempo y coste planteados a su inicio

28. ¿Se realiza una definición clara de la responsabilidad de cada miembro del equipo de desarrollo dentro de los proyectos de TI?

- ☐ Siempre
- ☐ Casi siempre
- ☐ Algunas veces
- ☐ Pocas veces
- ☐ Nunca

Compatibilidad con la metodología de desarrollo: Capacidad de adecuación o funcionamientos de los proyectos y las metodologías de desarrollo

29. ¿Los proyectos tienen concordancia con la metodología de desarrollo utilizada?

- ☐ Siempre
- ☐ Casi siempre
- ☐ Algunas veces
- ☐ Pocas veces
- ☐ Nunca

30. ¿Se realizan cambios en los proyectos con relación a la metodología de desarrollo utilizada?

- ☐ Nunca
- ☐ Pocas veces
- ☐ Algunas veces
- ☐ Casi siempre
- ☐ Siempre

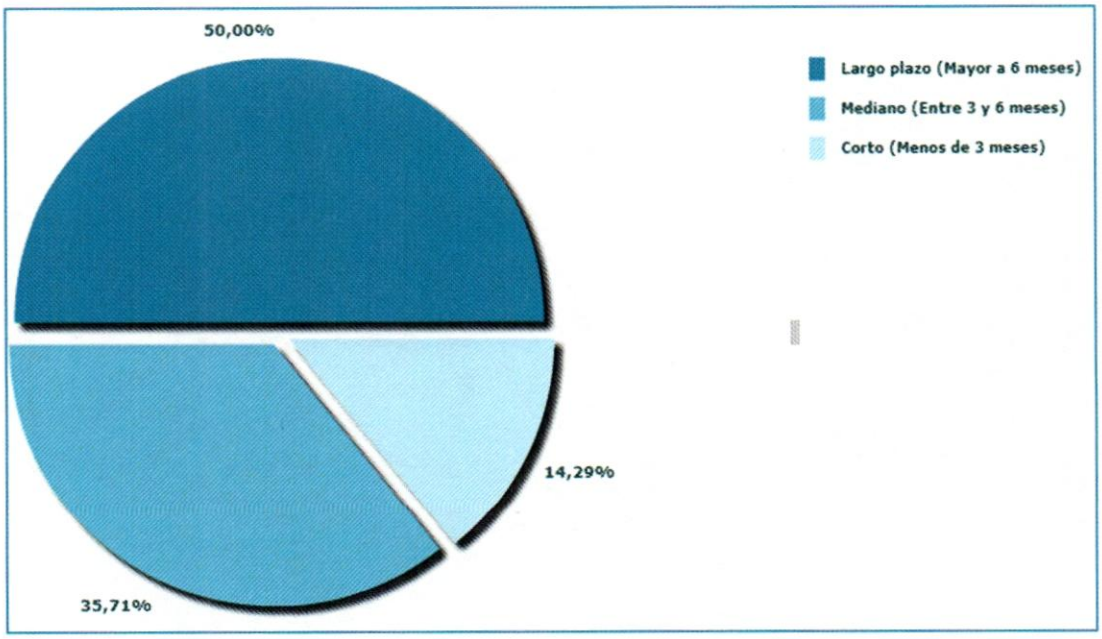
ANEXO C

RESULTADOS DEL CUESTIONARIO

1. Ha estado Usted involucrado en desarrollos de software a:

Tabla 12. Respuesta item 1

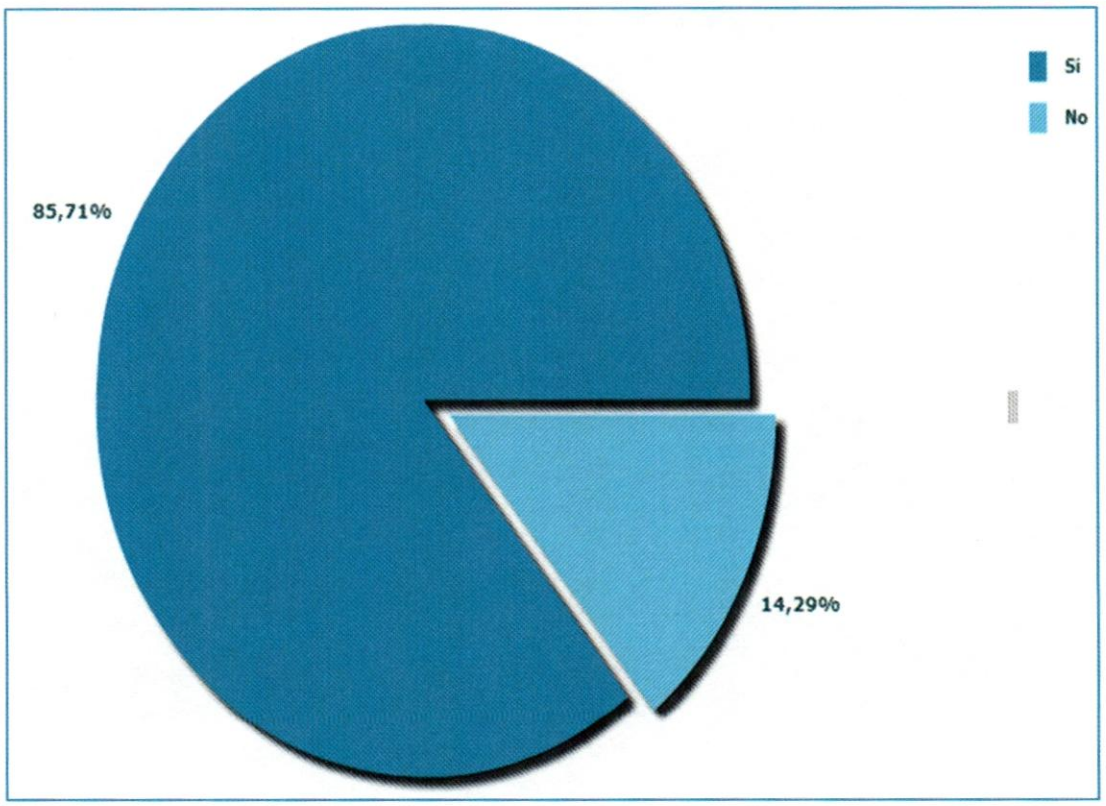
Respuesta	Cantidad	%
Largo plazo (Mayor a 6 meses)	7	50%
Mediano (Entre 3 y 6 meses)	5	36%
Corto (Menos de 3 meses)	2	14%
Total:	14	



2. ¿Existe una planificación de las actividades a desarrollar dentro del proyecto?

Tabla 13. Respuesta item 2

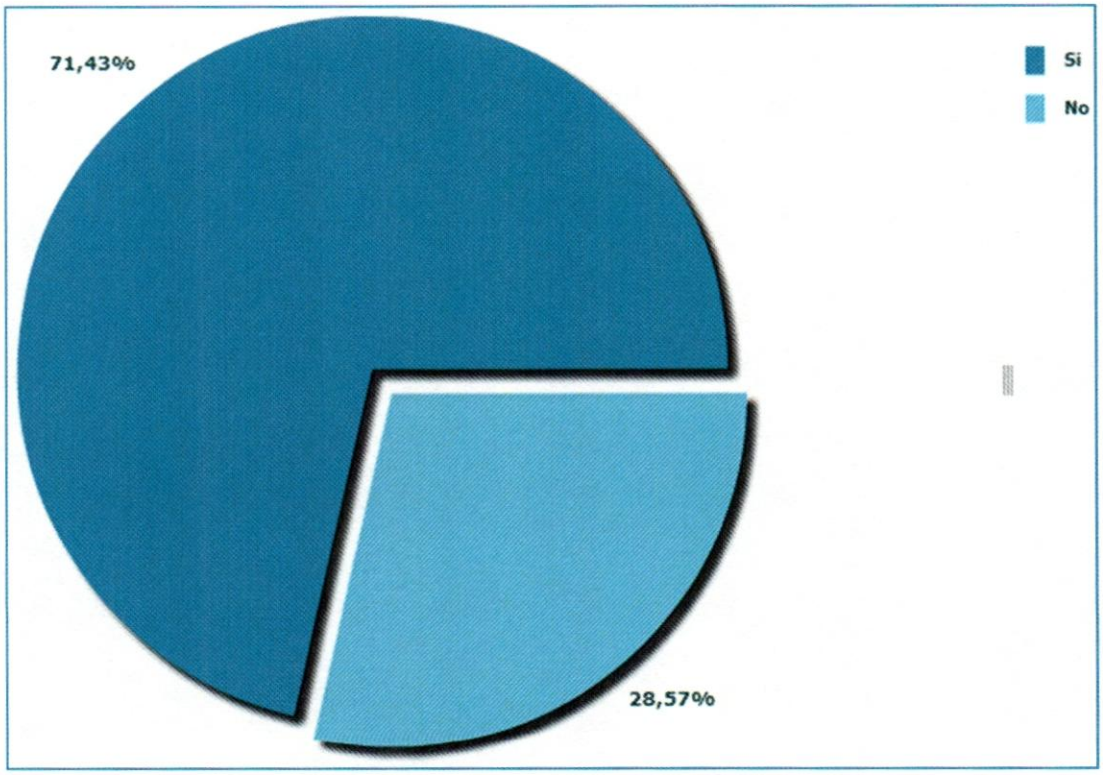
Respuesta	Cantidad	%
Si	12	86%
No	2	14%
Total:	14	



3. ¿Se define el tiempo de entrega para cada una de las actividades planificadas?

Tabla 14. Respuesta item 3

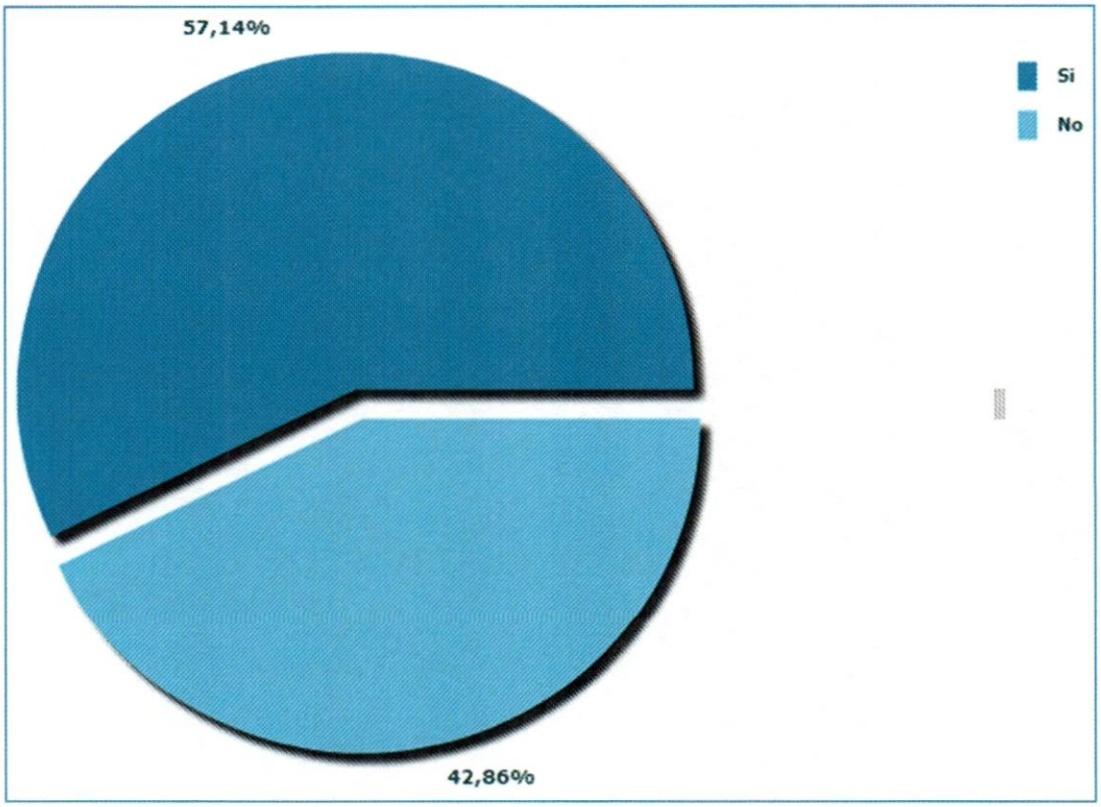
Respuesta	Cantidad	%
Si	10	71%
No	4	29%
Total:	14	



4. Una vez realizada las actividades de planificación y estimación, ¿Conoce Usted Cuál es la fecha estimada de finalización del proyecto o sistema a desarrollar?

Tabla 15. Respuesta item 4

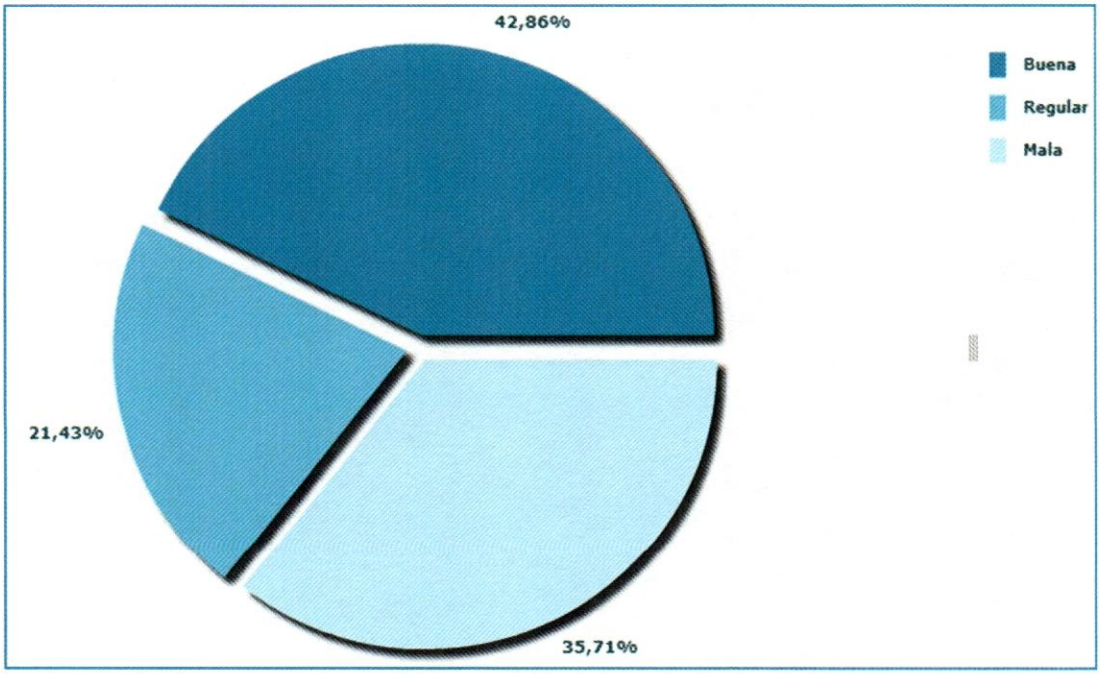
Respuesta	Cantidad	%
Si	8	57%
No	6	43%
Total:	**14**	



5. Documentar es una tarea tan necesaria e importante como el desarrollo del software. Considera Usted que la documentación entregada es:

Tabla 16. Respuesta item 5

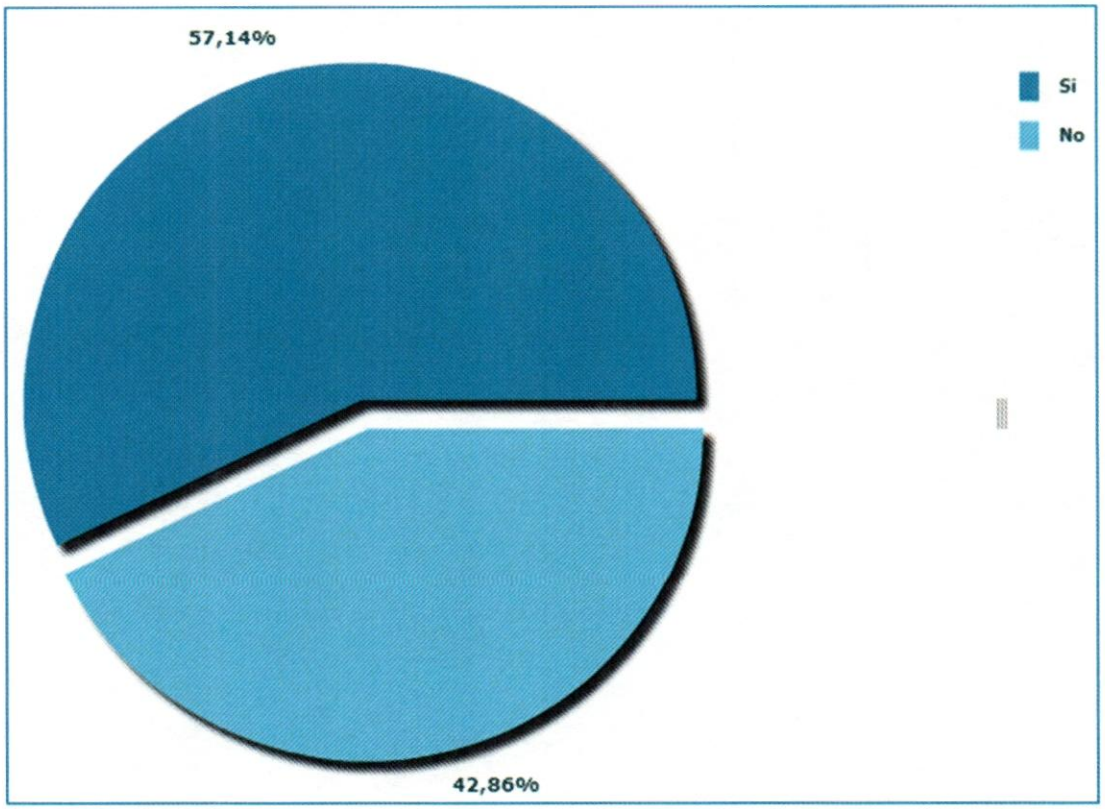
Respuesta	Cantidad	%
Buena	6	43%
Regular	3	21%
Mala	5	36%
Total:	**14**	



6. ¿Se realizan seguimientos de las tareas, los plazos, los costos y expectativas, con el fin de controlar los desarrollos de software, de tal forma que se eviten los retrasos en los proyectos?

Tabla 17. Respuesta item 6

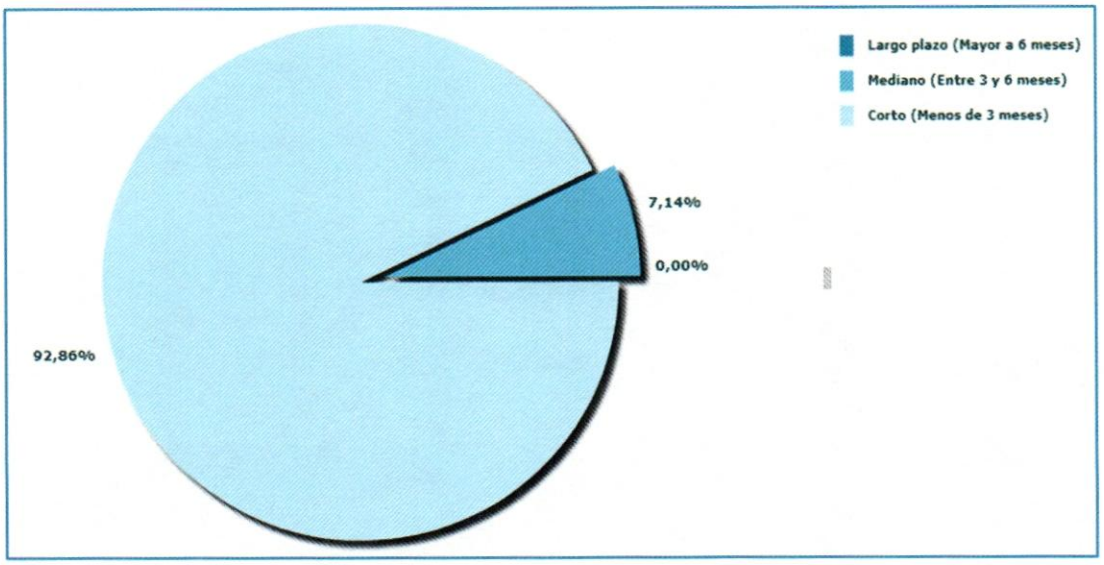
Respuesta	Cantidad	%
Si	8	57%
No	6	43%
Total:	**14**	



7. Los plazos de entrega de los primeros resultados se planifican a:

Tabla 18. Respuesta item 7

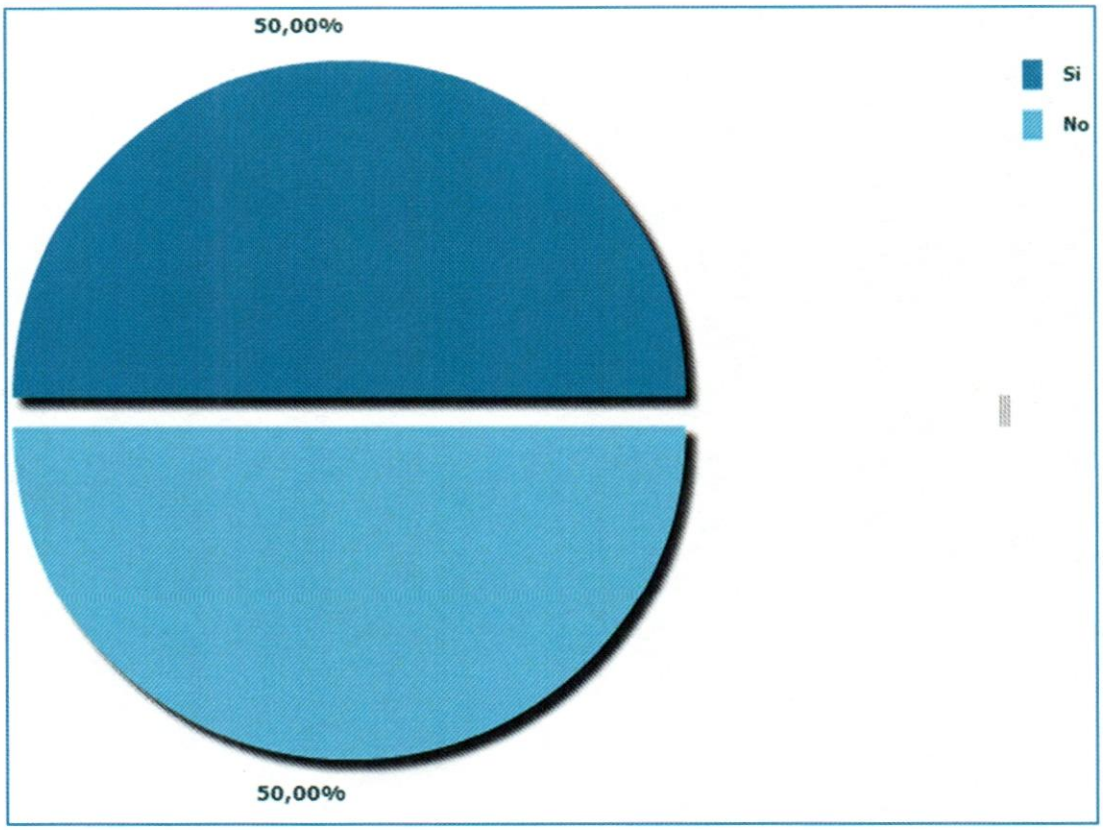
Respuesta	Cantidad	%
Largo plazo (Mayor a 6 meses)	0	0%
Mediano (Entre 3 y 6 meses)	1	7%
Corto (Menos de 3 meses)	13	93%
Total:	**14**	



8. ¿Los primeros resultados obtenidos del desarrollo de software, son medidos contra los planes?

Tabla 19. Respuesta item 8

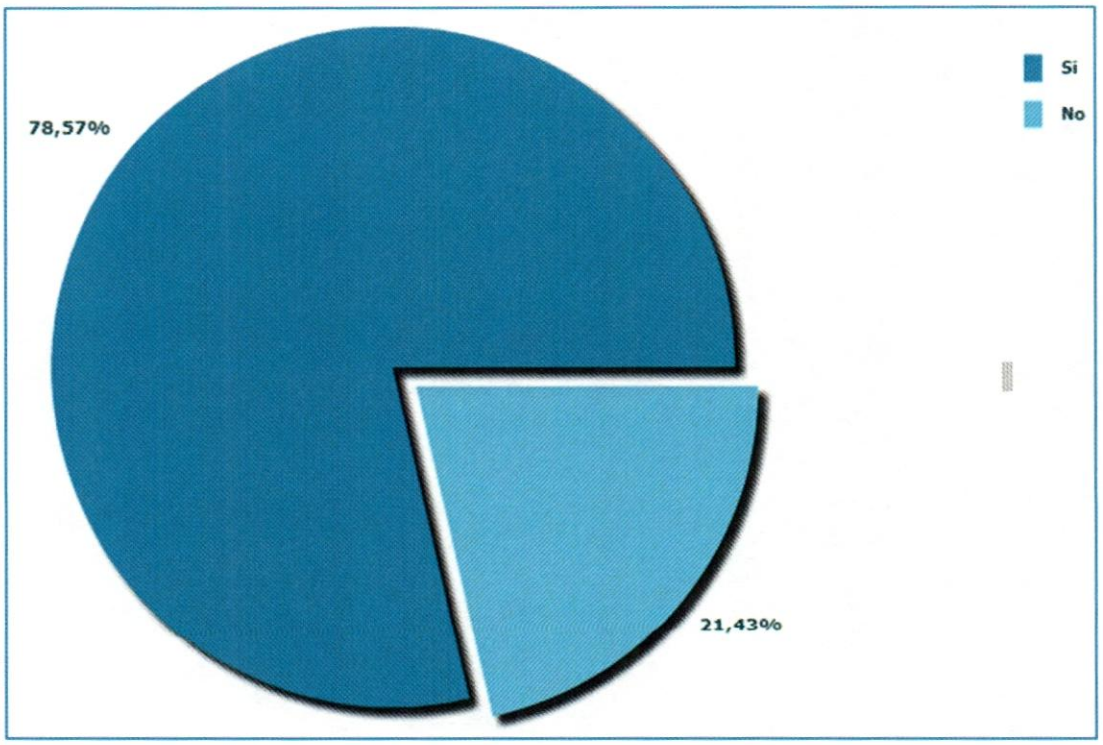
Respuesta	Cantidad	%
Si	7	50%
No	7	50%
Total:	**14**	



9. ¿Si existe discrepancia entre los resultados y los objetivos, se toman a tiempo las acciones correctivas?

Tabla 20. Respuesta item 9

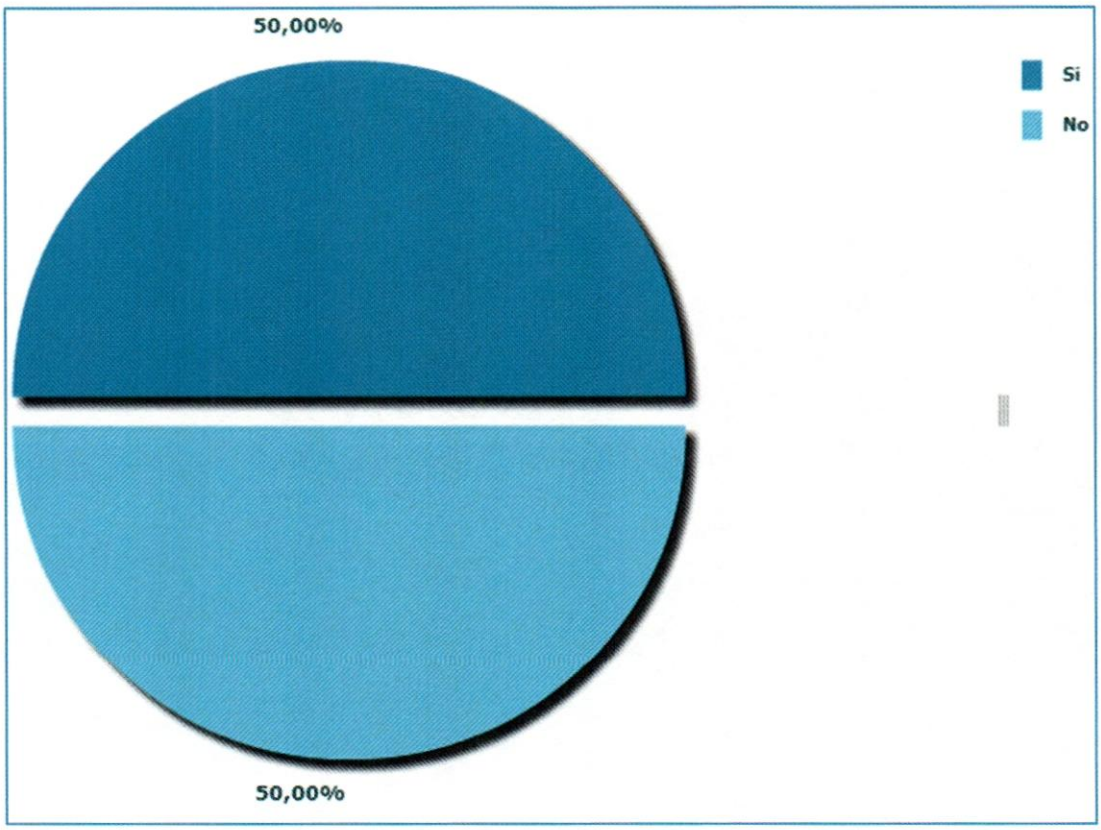
Respuesta	Cantidad	%
Si	11	79%
No	3	21%
Total:	**14**	



10. ¿Se definen procesos de control de cambios o ajuste de los objetivos?

Tabla 21. Respuesta item 10

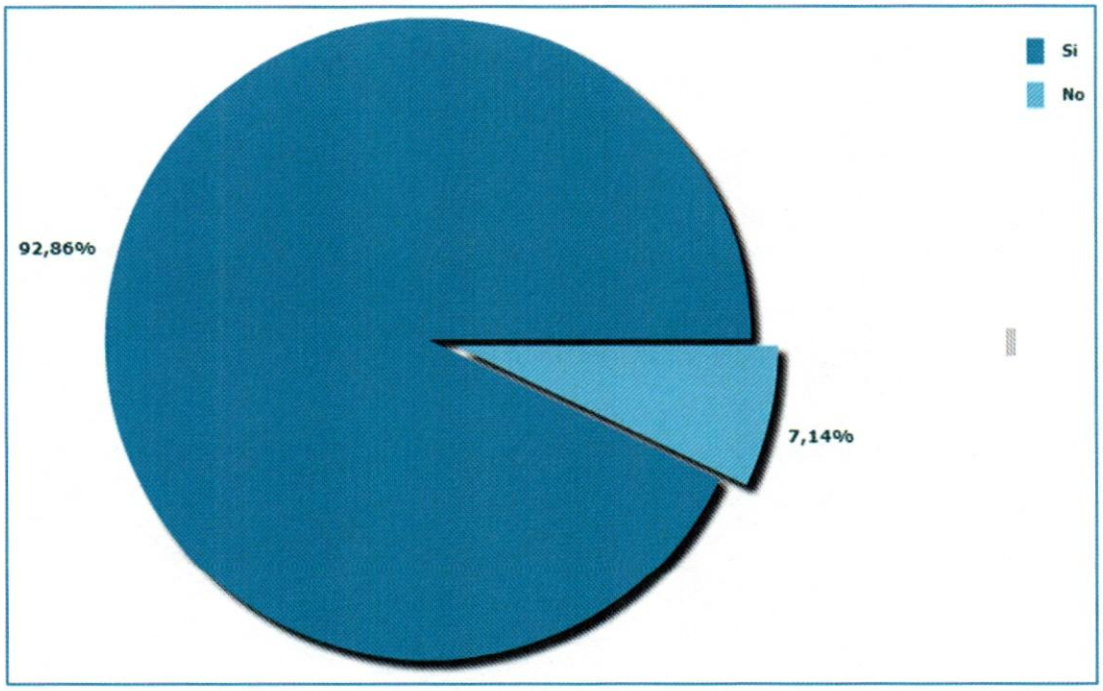
Respuesta	Cantidad	%
Si	7	50%
No	7	50%
Total:	**14**	



11. ¿Cuándo la gestión de los requisitos no se hace bien, aparecen altos niveles de re-trabajo a lo largo del proyecto?

Tabla 22. Respuesta item 11

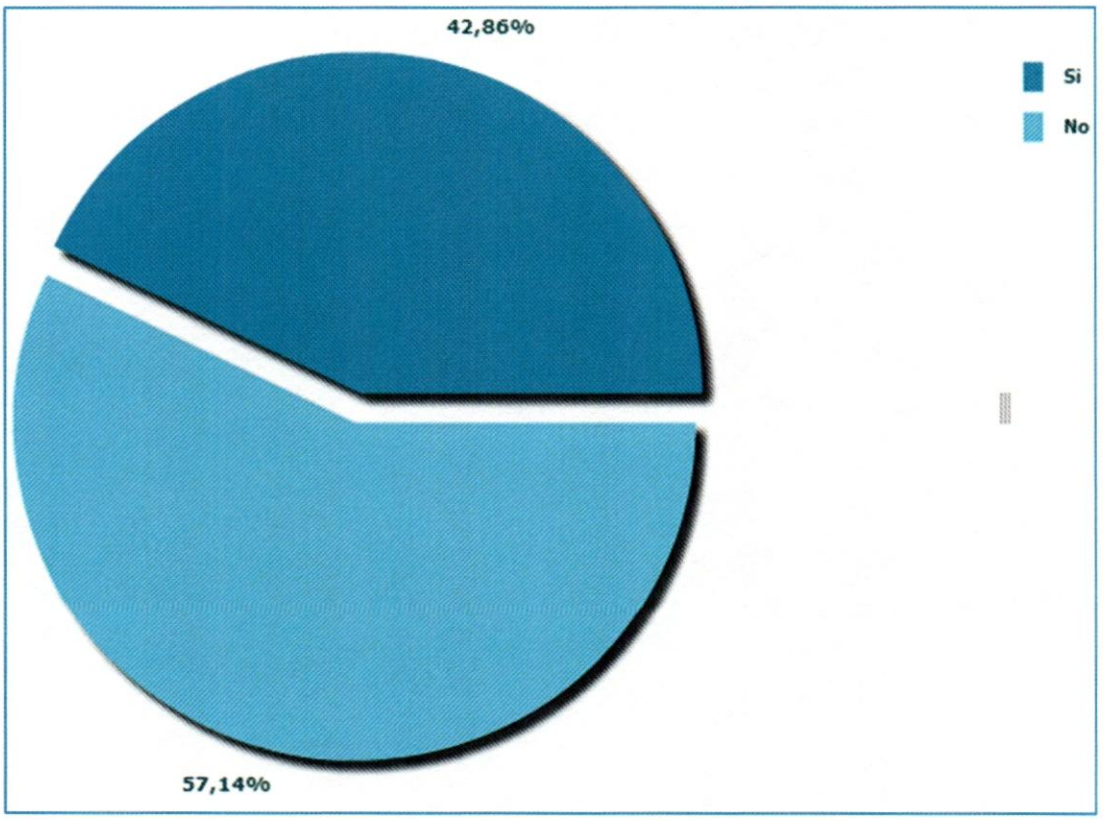
Respuesta	Cantidad	%
Si	13	93%
No	1	7%
Total:	**14**	



12. ¿Las actividades para el aseguramiento de calidad del software son detalladas en métricas de software para el control del proyecto, verificación y validación del software a lo largo del ciclo de vida?

Tabla 23. Respuesta item 12

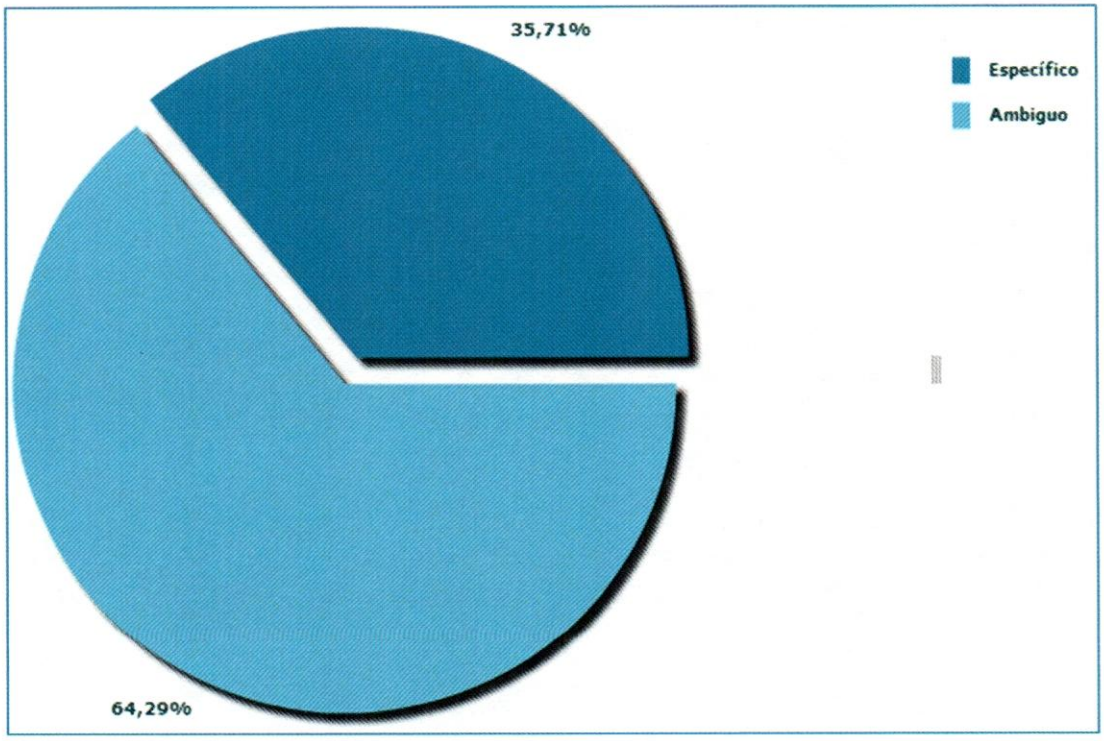
Respuesta	Cantidad	%
Si	6	43%
No	8	57%
Total:	**14**	



13. La definición del alcance del proyecto es:

Tabla 24. Respuesta item 13

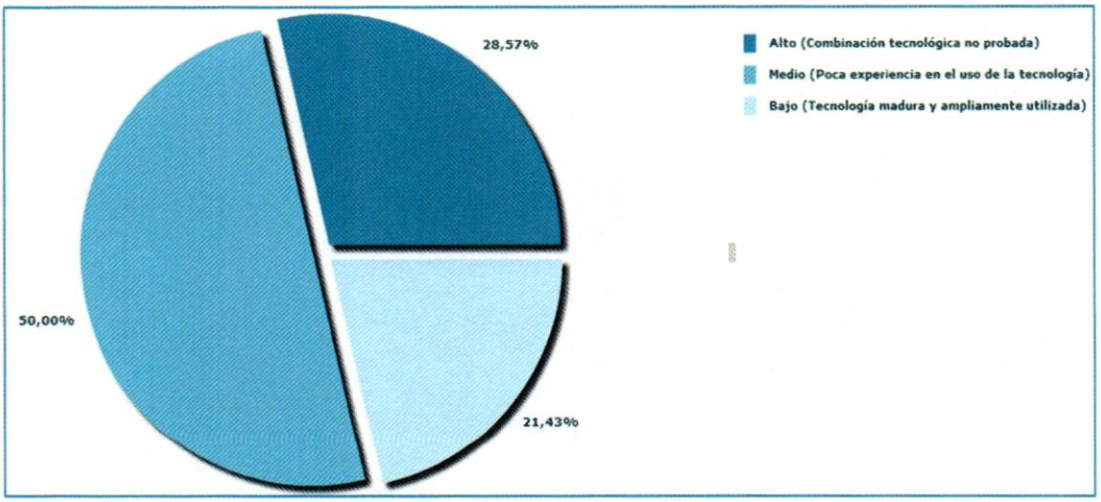
Respuesta	Cantidad	%
Específico	5	36%
Ambiguo	9	64%
Total:	**14**	



14. La combinación entre riesgo tecnológico y personal idóneo que componen la estrategia en la organización es:

Tabla 25. Respuesta item 14

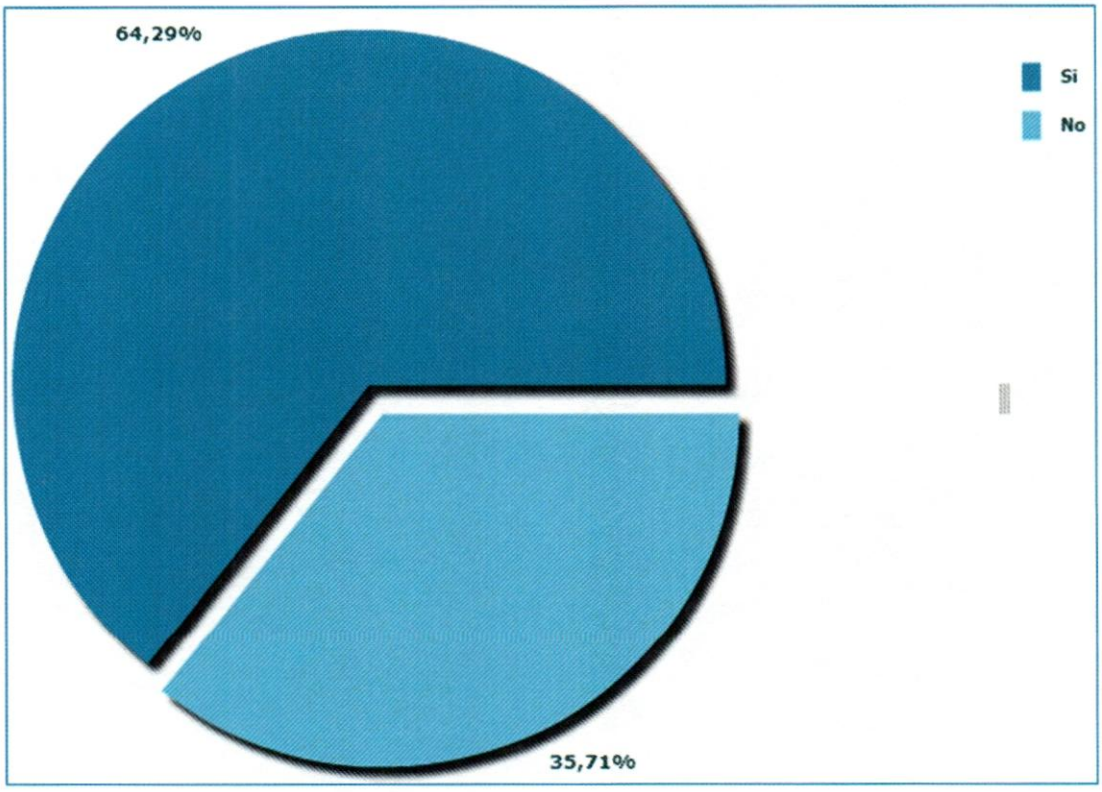
Respuesta	Cantidad	%
Alto (Combinación tecnológica no probada)	4	29%
Medio (Poca experiencia en el uso de la tecnología)	7	50%
Bajo (Tecnología madura y ampliamente utilizada)	3	21%
Total:	**14**	



15. Se presentan problemas con los clientes internos:

Tabla 26. Respuesta item 15

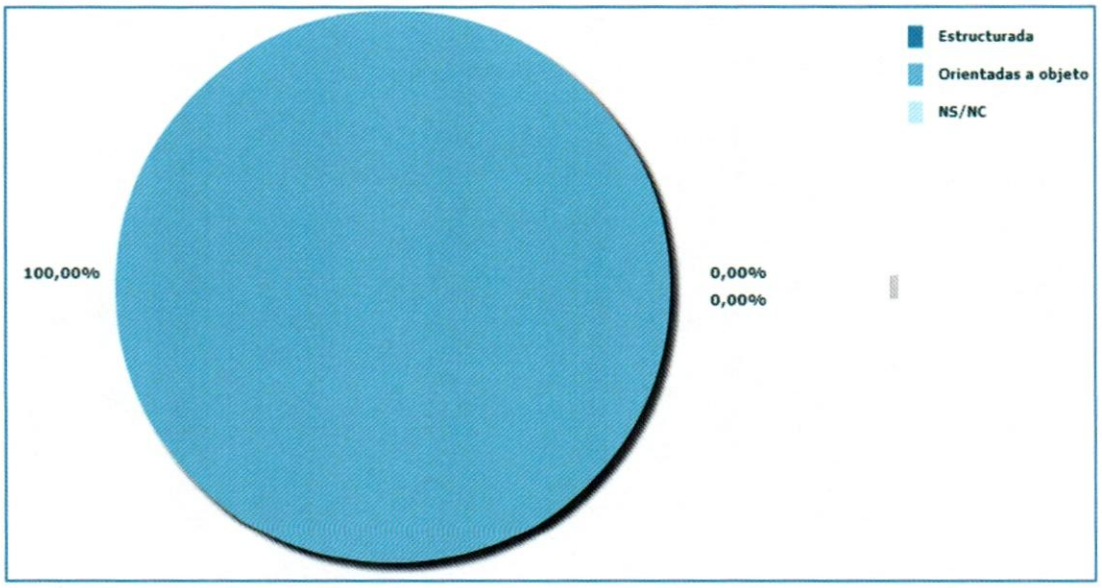
Respuesta	Cantidad	%
Si	9	64%
No	5	36%
Total:	**14**	



16. La metodología utilizada y aplicada para el desarrollo de diversos software es:

Tabla 27. Respuesta item 16

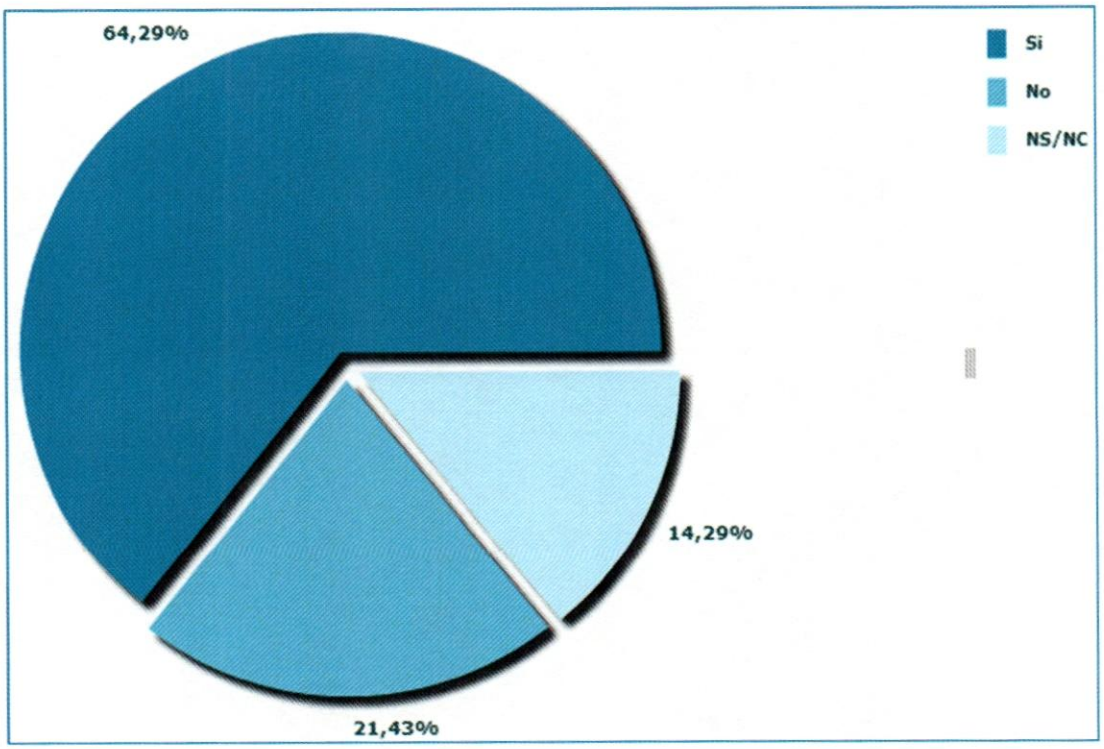
Respuesta	Cantidad	%
Estructurada	0	0%
Orientada a objetos	14	100%
NS/NC	0	0%
Total:	**14**	



17. La metodología utilizada es fácil aprender o enseñar

Tabla 28. Respuesta item 17

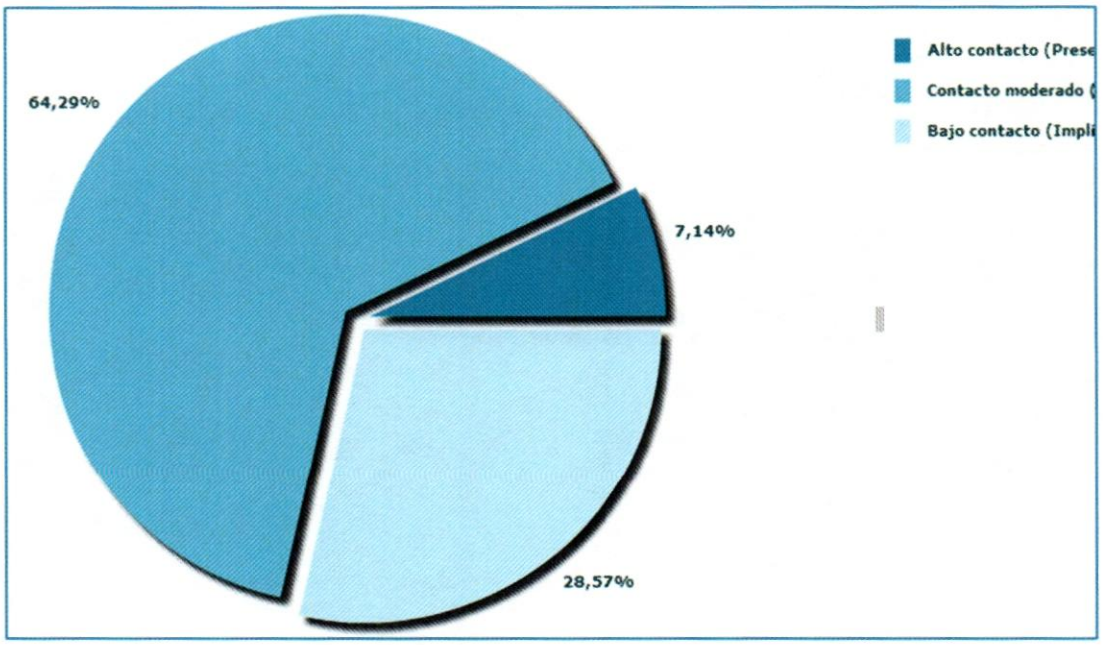
Respuesta	Cantidad	%
Si	9	64%
No	3	21%
NS/NC	2	14%
Total:	**14**	



18. Las acciones proporcionadas por los clientes para continuidad en las actividades en el desarrollo del software son:

Tabla 29. Respuesta item 18

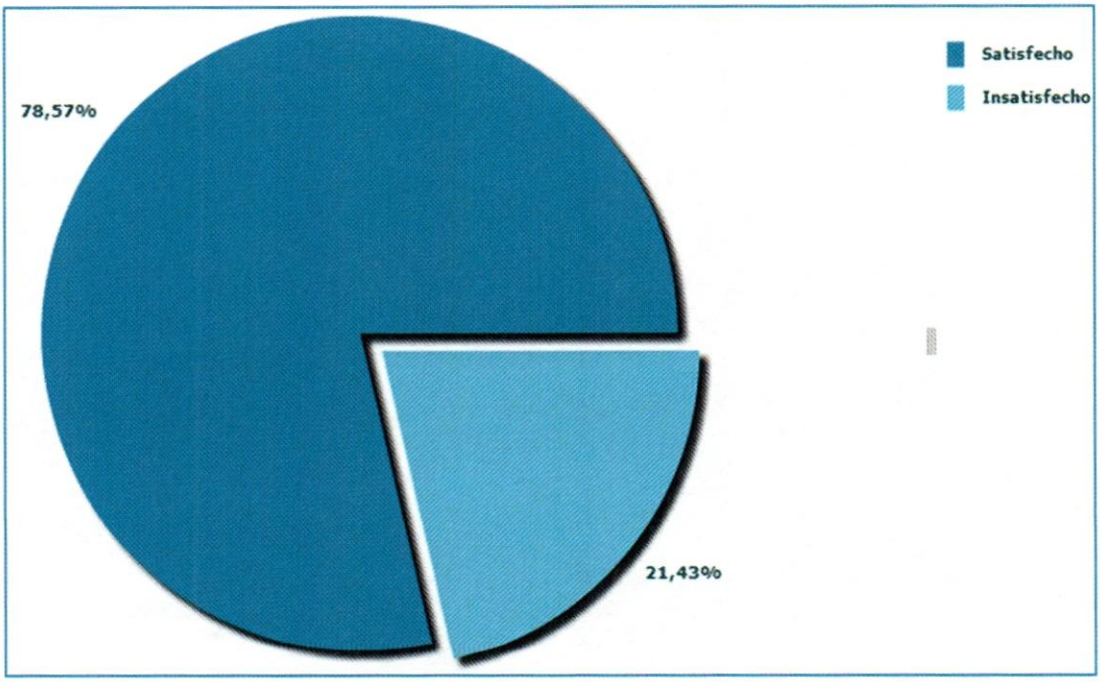
Respuesta	Cantidad	%
Alto contacto	1	7%
Contacto moderado	9	64%
Bajo contacto	4	29%
Total:	**14**	



19. Grado de satisfacción

Tabla 30. Respuesta item 19

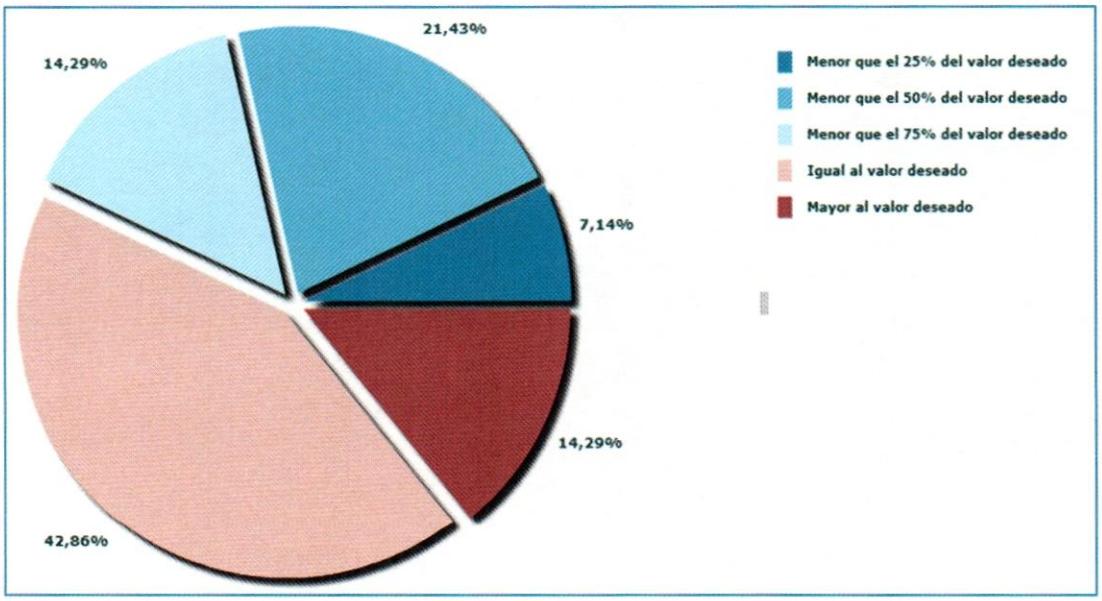
Respuesta	Cantidad	%
Satisfecho	11	79%
Insatisfecho	3	21%
Total:	**14**	



20. El valor de los costos de desarrollo o implementación de los proyectos son:

Tabla 31. Respuesta item 20

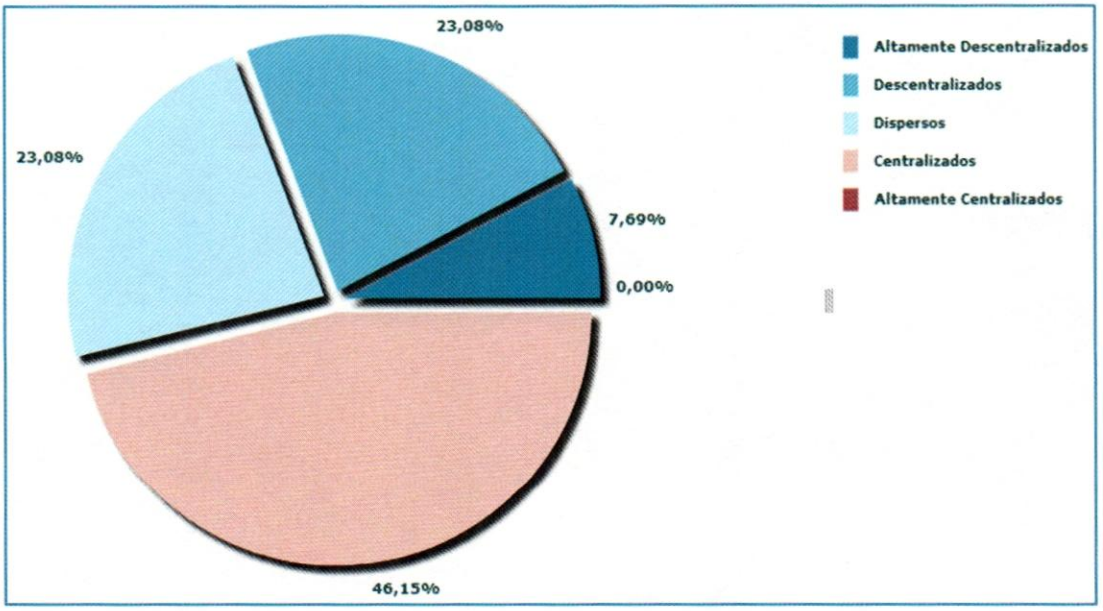
Respuesta	Cantidad	%
Menor que el 25% del valor deseado	1	7%
Menor que el 50% del valor deseado	3	21%
Menor que el 75% del valor deseado	2	14%
Igual al valor deseado	6	43%
Mayor al valor deseado	2	14%
Total:	**14**	



21. ¿Qué tan compleja es la estructura organizacional (organigrama)?

Tabla 32. Respuesta item 21

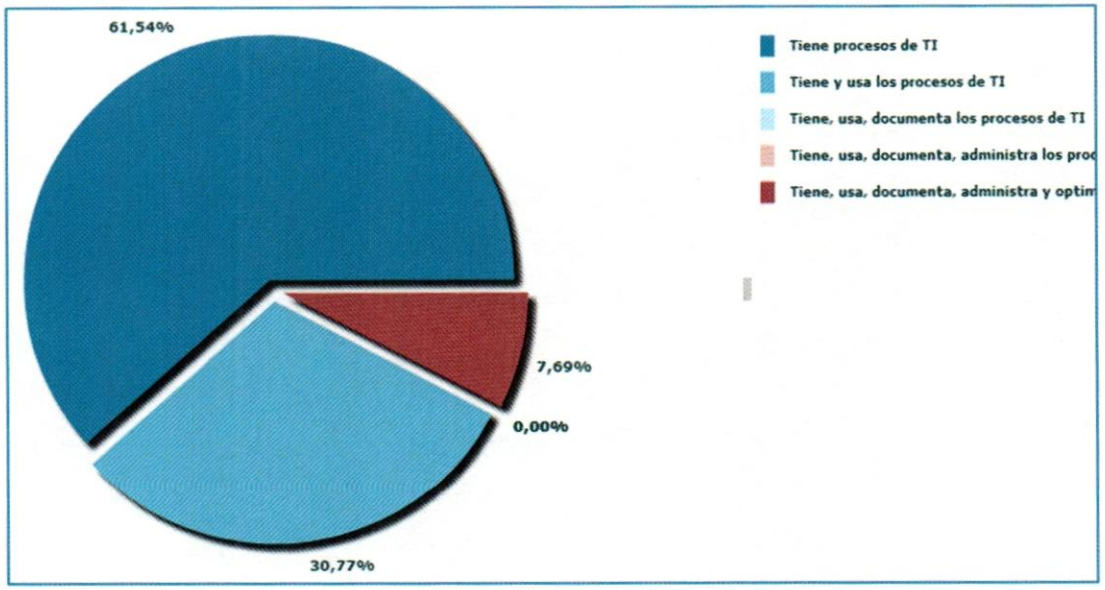
Respuesta	Cantidad	%
Altamente Descentralizados	1	7%
Descentralizados	3	21%
Dispersos	3	21%
Centralizados	7	50%
Altamente Centralizados	0	0%
Total:	**14**	



22. ¿Cuál es el estado de desarrollo de los procesos de negocios en la organización?

Tabla 33. Respuesta item 22

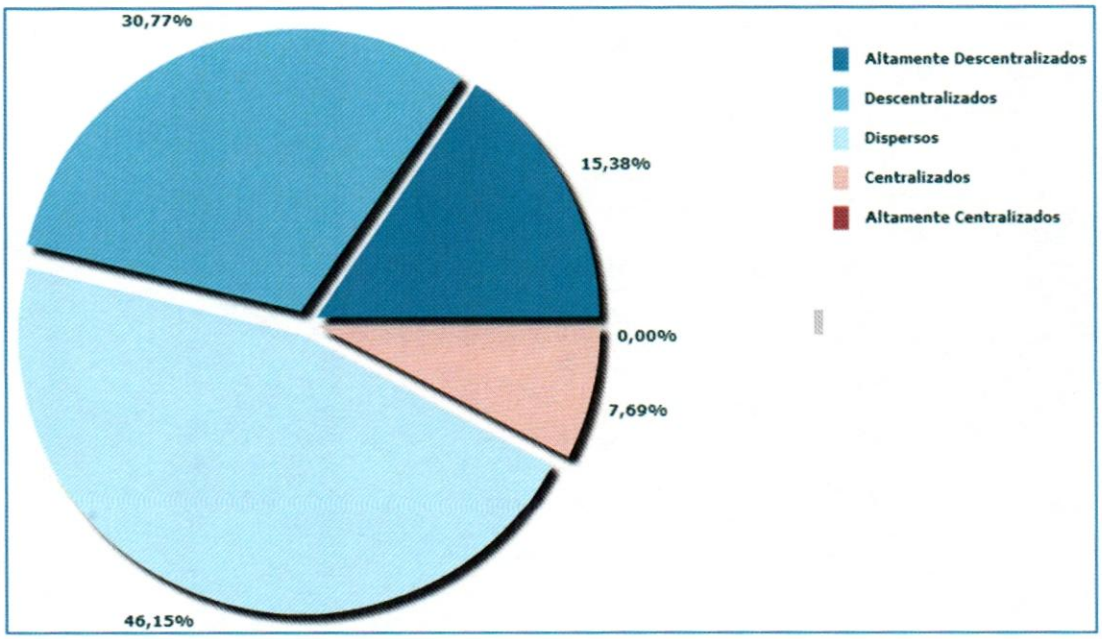
Respuesta	Cantidad	%
Tiene procesos de TI	9	64%
Tiene y usa los procesos de TI	4	29%
Tiene, usa, documenta los procesos de TI	0	0%
Tiene, usa, documenta, administra los procesos de TI	0	0%
Tiene, usa, documenta, administra y optimiza los procesos de TI	1	7%
Total:	**14**	



23. Dispersión de los usuarios

Tabla 34. Respuesta item 23

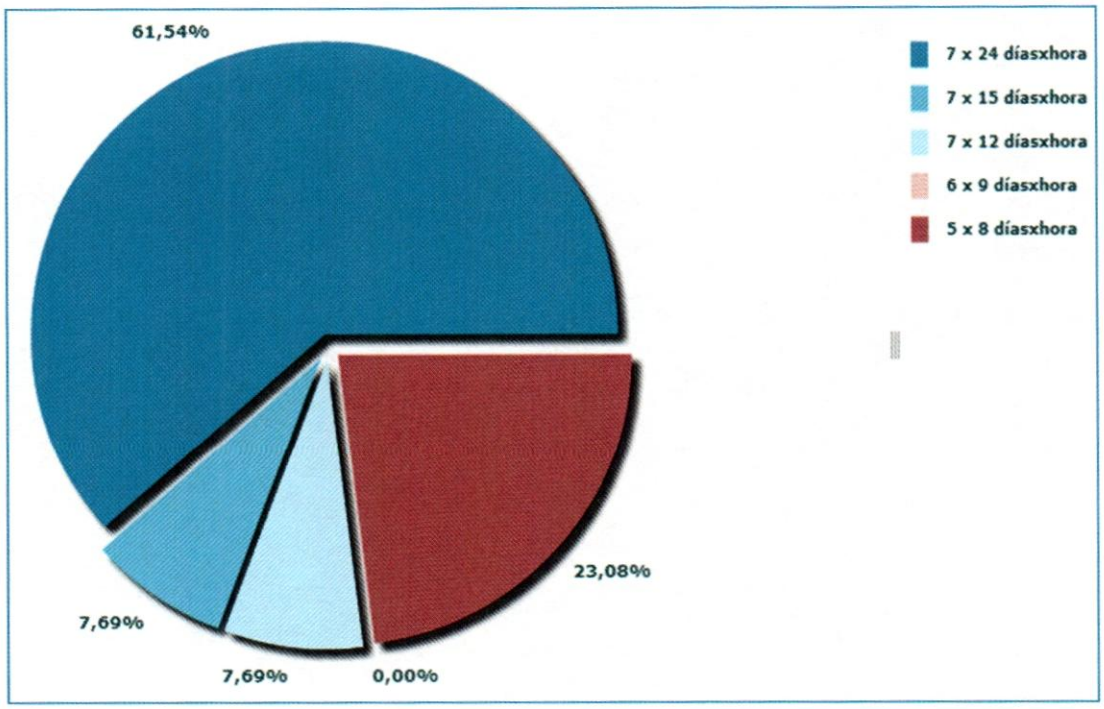
Respuesta	Cantidad	%
Altamente Descentralizados	2	14%
Descentralizados	4	29%
Dispersos	7	50%
Centralizados	1	7%
Altamente Centralizados	0	0%
Total:	**14**	



24. Los servicios de TI están disponibles y funcionan correctamente siempre que los clientes y usuarios deseen hacer uso de ellos, con una frecuencia de:

Tabla 35. Respuesta item 23

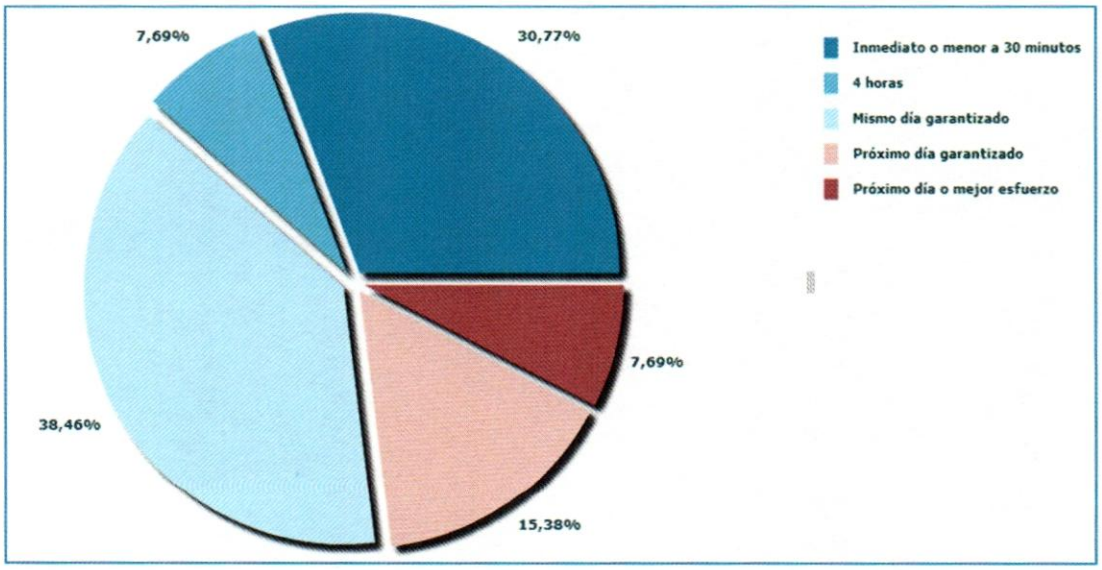
Respuesta	Cantidad	%
7 x 24 díasxhora	9	64%
7 x 15 díasxhora	1	7%
7 x 12 díasxhora	1	7%
6 x 9 díasxhora	0	0%
5 x 8 díasxhora	3	21%
Total:	**14**	



25. En los niveles de acuerdo de servicio (SLA), la provisión de los servicios acordados se establece en:

Tabla 36. Respuesta item 25

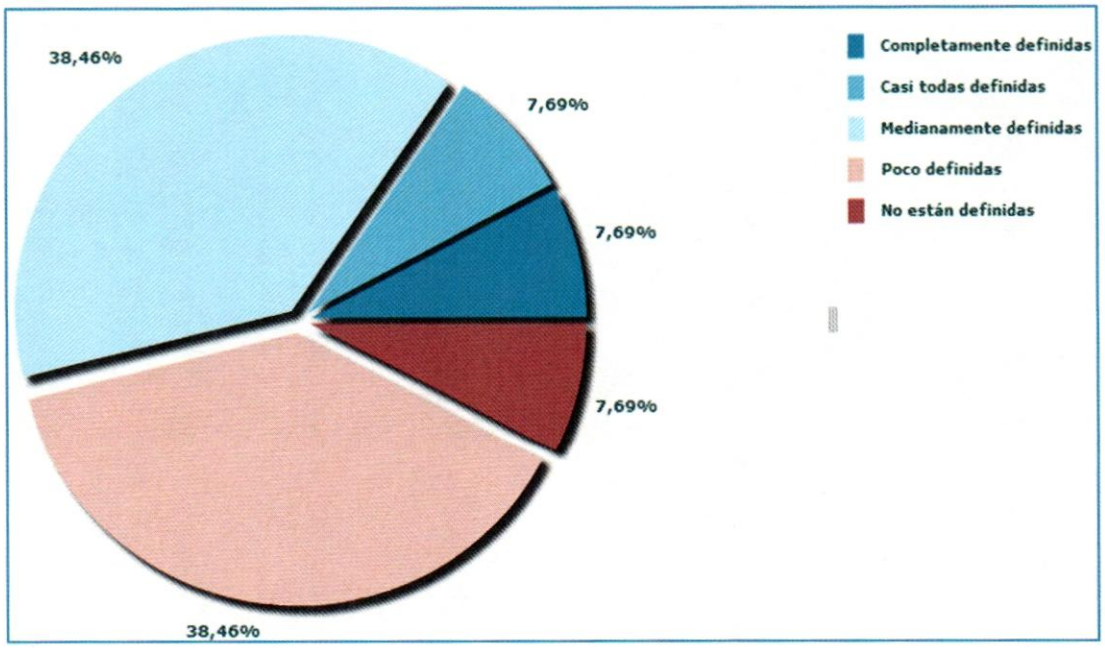
Respuesta	Cantidad	%
Inmediato o menor a 30 minutos	4	29%
4 horas	1	7%
Mismo día garantizado	6	43%
Próximo día garantizado	2	14%
Próximo día o mejor esfuerzo	1	7%
Total:	**14**	



26. Los planes para la renovación periódica de la infraestructura de TI están:

Tabla 37. Respuesta item 26

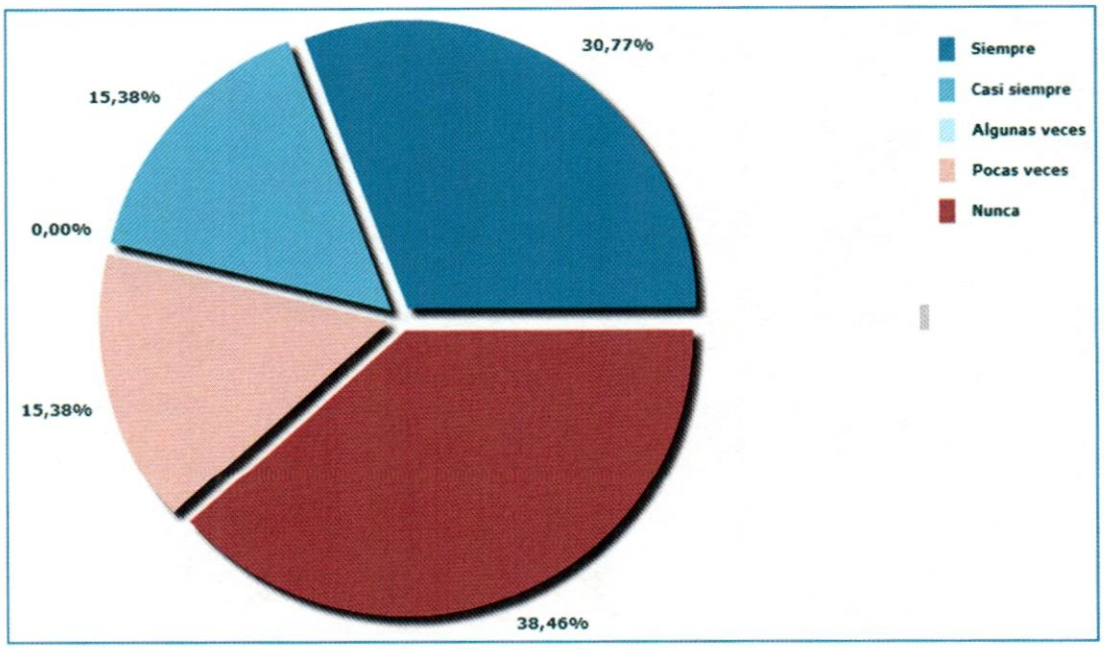
Respuesta	Cantidad	%
Completamente definidas	1	7%
Casi todas definidas	1	7%
Medianamente definidas	5	36%
Poco definidas	6	43%
No están definidas	1	7%
Total:	**14**	



27. ¿Se realizan los requerimientos de estudios de costo-beneficio para definir la factibilidad de los proyectos a ser desarrollados?

Tabla 38. Respuesta item 27

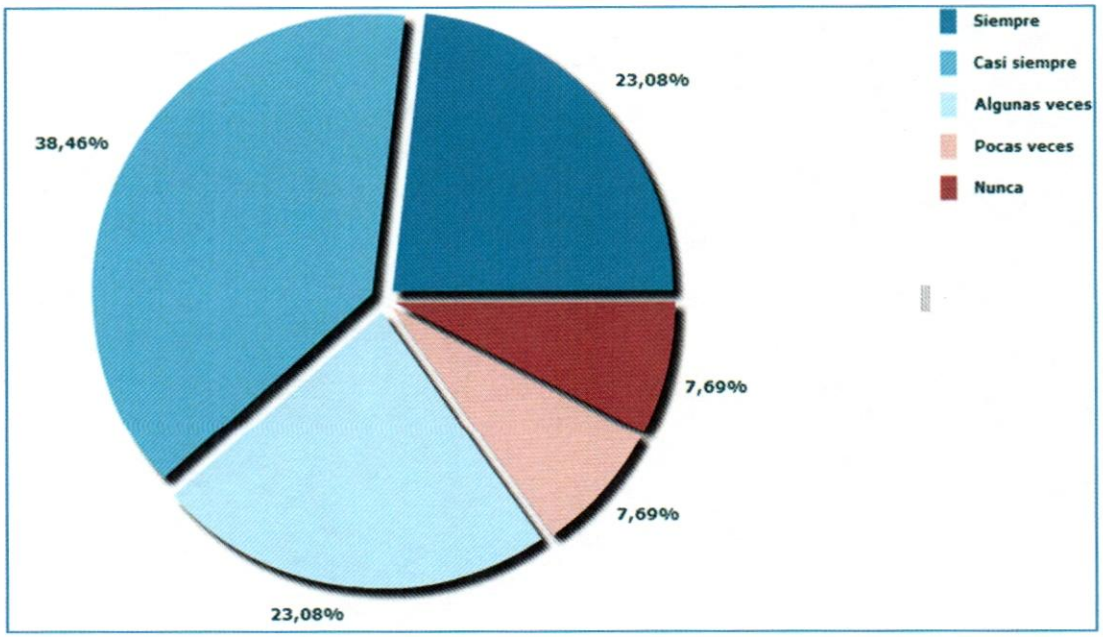
Respuesta	Cantidad	%
Siempre	4	29%
Casi siempre	2	14%
Algunas veces	0	0%
Pocas veces	2	14%
Nunca	6	43%
Total:	**14**	



28. ¿Se realiza una definición clara de la responsabilidad de cada miembro del equipo de desarrollo dentro de los proyectos de TI?

Tabla 39. Respuesta item 28

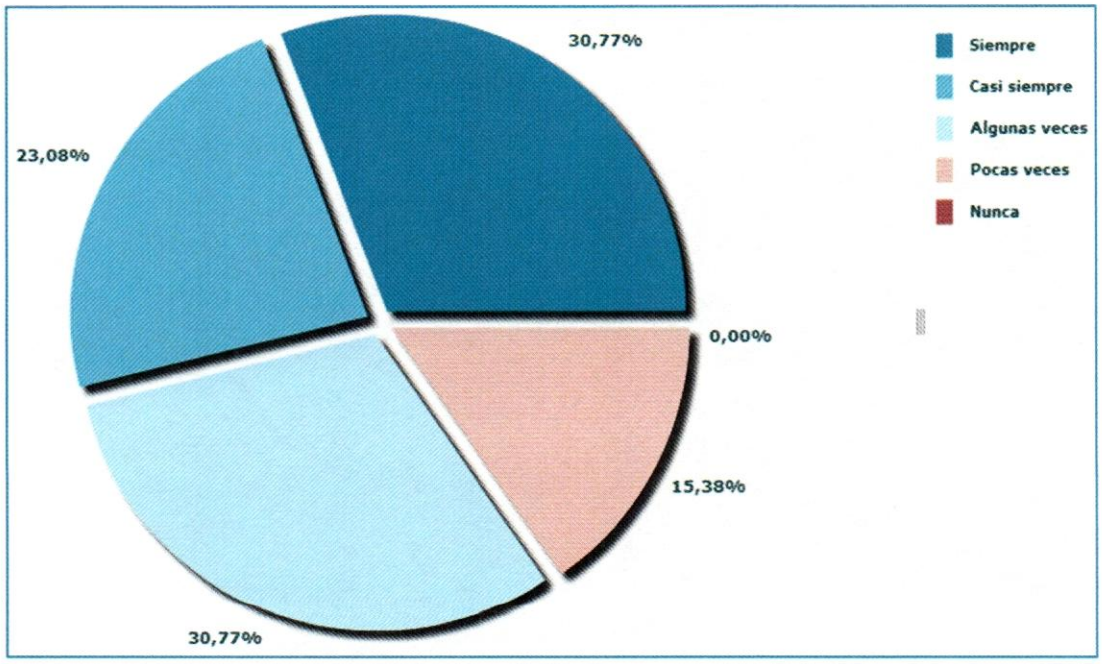
Respuesta	Cantidad	%
Siempre	3	21%
Casi siempre	6	43%
Algunas veces	3	21%
Pocas veces	1	7%
Nunca	1	7%
Total:	**14**	



29. ¿Los proyectos tienen concordancia con la metodología de desarrollo utilizada?

Tabla 40. Respuesta item 29

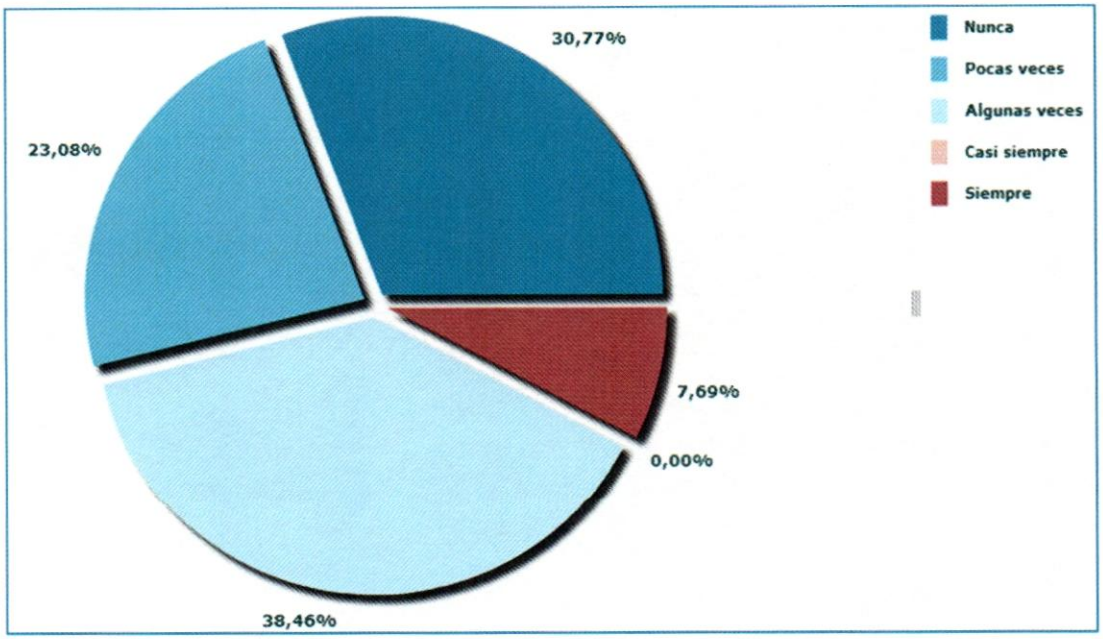
Respuesta	Cantidad	%
Siempre	4	29%
Casi siempre	3	21%
Algunas veces	5	36%
Pocas veces	2	14%
Nunca	0	0%
Total:	**14**	



30. ¿Se realizan cambios en los proyectos con relación a la metodología de desarrollo utilizada?

Tabla 41. Respuesta item 30

Respuesta	Cantidad	%
Nunca	4	29%
Pocas veces	3	21%
Algunas veces	6	43%
Casi siempre	0	0%
Siempre	1	7%
Total:	**14**	



ANEXO D

CONSTANCIA DE VALIDACIÓN DEL CUESTIONARIO

CONSTANCIA DE VALIDACIÓN

Yo, PEDRO BONILLO, titular de la cédula de identidad N°. 10.868.538, de profesión DR. CIENCIAS DE LA COMPUTACIÓN, ejerciendo actualmente el cargo de PROFESOR, en la institución UCAB (POSTGRADO).

Por medio de la presente, hago constar que he revisado y validado a los fines de su aplicación el instrumento (cuestionario) para el trabajo de grado titulado: "Modelo de madurez para la selección de una metodología de desarrollo de software en la Administración Pública Nacional".

En Caracas, a los 15 días del mes de ABRIL del 2014.

Bonillo Rangel Pedro

Firma

ANEXO E

ACEPTACIÓN DE LA EMPRESA

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para Industrias

juventud BICENTENARIA

Caracas, 27 de marzo de 2.014

Ciudadana
Lic. Luisana Castillo Gutiérrez
Presente.-

Estimada Lic. Castillo

Me complace dirigirme a usted en la oportunidad de enviarle un respetuoso saludo y a su vez informarle por medio de la presente que formalizamos la decisión de la Oficina de Tecnología y Sistemas del Ministerio del Poder Popular para Industrias de aceptar su solicitud, prestando el apoyo y la colaboración necesaria para las actividades relacionadas al estudio de investigación: ***"MODELO DE MADUREZ PARA LA SELECCIÓN DE UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE PARA LA ADMINISTRACIÓN PÚBLICA NACIONAL"***, correspondiente al trabajo de grado para optar al título de **Magister en Sistemas de Información en la Universidad Católica Andrés Bello**.

Sin más a que hacer referencia, se despide cordialmente,

REPÚBLICA BOLIVARIANA DE VENEZUELA
Gobierno Bolivariano de Venezuela
Ministerio del Poder Popular para Industrias
DIRECCIÓN DE LA OFICINA DE TECNOLOGÍA Y SISTEMAS

Angelo Bitorzoli H.
Director General (E) de la Oficina de Tecnología ySistemas
Ministerio del Poder Popular para Industrias

Dirección: Av. Urdaneta. Esquina de Pelota a Ibarras, Edificio Central. Piso 1.
Teléfono: (0212) 596-7608