



**FACULTAD DE TELEMÁTICA**

**MODELO DE CALIDAD PARA LA MICROEMPRESA  
BASADO EN MOPROSOFT**

TESIS  
QUE PARA OBTENER EL GRADO DE  
MAESTRO EN COMPUTACIÓN

Presenta:

**SERGIO ALAN FLORES ROSALES**

Asesores:

Dr. C. Nicandro Farías Mendoza  
M. en C. Armando Román Gallardo

**COLIMA, COL., FEBRERO DE 2012**

Colima, Col., Diciembre de 2011

M. en C. Ricardo Acosta Díaz  
Director de la Facultad de Telemática

P R E S E N T E

Por medio de este conducto informo que le trabajo de tesis para obtener el grado de Maestría en Computación del C. SERGIO ALAN FLORES ROSALES, ha sido revisado y aprobado por su servidor y por el M.C. ARMANDO ROMÁN GALLARDO y puede continuar con los trámites de titulación que estén estipulados.

El título de la tesis es: "MODELO DE CALIDAD PARA LA MICROEMPRESA BASADO EN MOPROSOFT", cuyo contenido ha sido examinado y encontrado adecuado a la finalidad que se propone conseguir y ha quedado desarrollado bajo los siguientes puntos:

**Capítulo I.** Introducción

**Capítulo II.** Estado del arte

**Capítulo III.** Modelo de calidad para la microempresa basado en MoproSoft

**Capítulo IV.** Especificación de procesos conforme a MoproSoft

**Capítulo V.** Pruebas y Resultados

**Capítulo VI.** Conclusiones

**Referencias**

Sin otro particular, reciba un cordial saludo.



Dr. Nicandro Farías Mendoza

Nombre de Profesor  
Nombramiento  
Institución  
Asesor(a) de la Tesis

Atte.



M.C. Armando Román Gallardo

Nombre de Profesor  
Nombramiento  
Institución  
Coasesor(a) de la Tesis

## **RESUMEN**

En México, las Micro, Pequeñas y Medianas empresas emplean alrededor del 78% del total de la población económicamente activa (Vázquez, 2002), es por ello la importancia de apoyar su desarrollo. Los estándares de calidad ayudan a hacerlas competentes en el mercado laboral.

Dichos estándares se definen por organizaciones internacionales y se diseñan para empresas medianas o grandes. El problema de adaptar los estándares de calidad no existe solamente en nuestro país, en Europa hay estudios que muestran las mismas tendencias (Oktaba, 2008).

La solución que se presenta en este documento, es un modelo basado en el Modelo de Procesos de Software que cumpla con las necesidades de las microempresas, MoproSoft cuenta con tres niveles, en el Nivel de Alta Dirección y de Gerencia, se propone unir en un solo nivel, en el Nivel de Operación se conserva la administración de proyectos específicos y el desarrollo y mantenimiento de software cambia.

## **ABSTRACT**

In Mexico, the Micro, Small and Medium Enterprises use about 78% of the total economically active population (Vazquez, 2002), which is why the importance of supporting their development. Quality standards help them competent in the job market.

These standards are defined by international organizations and are designed for medium or large enterprises. The problem of matching the quality standards does not exist only in our country, in Europe there are studies that show the same trends (Oktaba, 2008).

The solution presented in this paper, is a model based on the Software Process Model that meets the needs of micro enterprises, MoproSoft has three levels, on the Senior Management level and Management level the propose is join both levels in a single level, at the Operating Level is kept the specific project management and in software development and maintenance are changed.

A mis padres por apoyarme para seguir con mis estudios y motivarme a superarme académicamente y en la vida.

A mis hermanas por su dedicación al estudio que me motiva a superarme y tratar de ser un ejemplo para ellas.

A mis tíos Carlos y Amalia por estar siempre en disposición de ayudarme y apoyarme cuando lo necesito.

A la Universidad de Colima por crear el programa de Posgrado que me permitió seguir mis estudios e inclusive presentar mi trabajo en congresos gracias a los apoyos con los que conté.

# ÍNDICE

	Pág.
<b>CAPÍTULO I. INTRODUCCIÓN</b>	<b>1</b>
1.1 Contexto del problema	1
1.2 Antecedentes del problema a investigar	2
1.3 Síntesis de trabajos relacionados	3
1.4 Planteamiento del problema	6
1.5 Hipótesis del trabajo	7
1.6 Objetivos	7
1.7 Metodologías Empleadas	8
1.8 Organización del documento	10
<b>CAPÍTULO II. ESTADO DEL ARTE</b>	<b>12</b>
2.1 Marco histórico	12
2.2 Marco contextual	13
2.3 Marco teórico	15
<b>CAPITULO III MODELO DE CALIDAD PARA LA MICROEMPRESA BASADO EN MOPROSOFT</b>	<b>21</b>
3.1 Gerencia	29
3.1.1 Gestión de proyectos	29
3.1.2 Gestión de procesos	34
3.1.3 Gestión de recursos	36
3.2 Operación	38
3.2.1 Fase de Exploración	39
3.2.2 Fase de Planificación	42

3.2.3 Fase de Diseño	44
3.2.4 Fase de Producción	47
3.2.5 Fase de Implantación	50

## **CAPÍTULO IV ESPECIFICACIÓN DE PROCESOS CONFORME A MOPROSOFT**

**53**

4.1 Especificación de procesos para la categoría de Gerencia	53
4.1.1 Gestión de Proyectos	53
4.1.2 Gestión de Procesos	57
4.1.3 Gestión de Recursos	63
4.2 Especificación de procesos para la categoría de Operación	63
4.2.1 Fase de Exploración	63
4.2.2 Fase de Planificación	68
4.2.3 Fase de Diseño	70
4.2.4 Fase de Implantación	72

## **CAPITULO V PRUEBAS Y RESULTADOS**

**77**

## **CAPÍTULO VI CONCLUSIONES**

**91**

6.1 Discusión	91
6.2 Cumplimiento de las hipótesis de trabajo	91
6.3 Recomendaciones	92

## **REFERENCIAS**

**93**

## ÍNDICE DE FIGURAS

	Pág.
Figura 1 Modelos de calidad de software utilizados	6
Figura 2 Impacto de Metodologías ágiles en la calidad	17
Figura 3 Número de personal en elaboración de Software	18
Figura 4 Metodologías utilizadas	19
Figura 5 Documentación de calidad	19
Figura 6 Modelo de calidad de Software	20
Figura 7 Alta dirección de MoproSoft	21
Figura 8 Gestión de proyectos y procesos en MoproSoft	22
Figura 9 Gestión de recursos en MoproSoft	24
Figura 10 Administración de proyectos específicos en MoproSoft	25
Figura 11 Desarrollo y mantenimiento de Software en MoproSoft	26
Figura 12 Modelo de calidad para la microempresa basado en MoproSoft	28
Figura 13 Nivel Gerencia	34
Figura 14 Gestión de Proyectos	30
Figura 15 Gestión de Procesos	34
Figura 16 Gestión de Recursos	37
Figura 17 Nivel Operación	38
Figura 18 Fase de Exploración	39
Figura 19 Fase de Planificación	43
Figura 20 Fase de Diseño	44
Figura 21 Fase de Producción	47
Figura 22 Desarrollo en Scrum	48
Figura 23 Fase de Implantación	50
Figura 24 Anverso de tarjeta de usuario	65
Figura 25 Reverso de tarjeta de usuario	65
Figura 26 Tarjeta CRC 1	70
Figura 27 Tarjeta CRC 2	71



Figura 28 Tarjeta CRC 3	71
Figura 29 Tarjeta CRC 4	71
Figura 30 Ciclo de vida	83

## ÍNDICE DE TABLAS

Tabla 1 Comparativa entre MoproSoft y el modelo propuesto	27
Tabla 2 Mantenimiento del Proyecto	55
Tabla 3 Calendario de Procesos	57
Tabla 4 Definición de escalas de impacto	59
Tabla 5 Otros impactos de riesgo en el proyecto	60
Tabla 6 Desarrollo del Proyecto	73
Tabla 7 Fases de XP	73
Tabla 8 Fases DSDM	73
Tabla 9 Metodología propuesta	74
Tabla 10 Proyecto MoproSoft	75
Tabla 11 Proyecto Modelo propuesto	77
Tabla 12 Definición de la estructura de equipo de trabajo	82
Tabla 13 Interfaces externas	83
Tabla 14 Estimación de esfuerzo del proyecto	84
Tabla 15 Protocolo de entrega	85
Tabla 16 Entrega de documentos	85
Tabla 17 Tiempos del nivel Operación	87
Tabla 18 Entregables del sistema	88

## **CAPÍTULO I. INTRODUCCIÓN**

En México, las Micro, Pequeñas y Medianas empresas (MiPyMES) emplean alrededor del 78% del total de la población económicamente activa (Vázquez, 2002), es por ello la importancia de apoyar su desarrollo. Los estándares de calidad ayudan a hacerlas competentes en el mercado laboral.

En este capítulo se mencionan algunos problemas del porqué se hace necesario aplicar estándares de calidad en microempresas desarrolladoras de Software. Dichos estándares se definen por organizaciones internacionales y se diseñan para empresas medianas o grandes. El problema de adaptar los estándares de calidad no existe solamente en nuestro país, en Europa hay estudios que muestran las mismas tendencias y en donde las microempresas agrupan alrededor del 30% de la fuerza laboral de esa área y en número de empresas ocupan 78% (Oktaba, 2008).

Uno de los problemas a resolver es adaptar dichos estándares a microempresas y la solución que se presenta en este documento, es crear un modelo basado en MoproSoft que cumpla con las necesidades de éstas. Además, se mencionan algunos trabajos enfocados en resolver el problema de las microempresas al adaptar modelos de calidad y otros que mencionan problemas laborales de las microempresas desarrolladoras de Software.

### **1.1 Contexto del problema**

Los estándares para el aseguramiento de la calidad del Software son definidos por organizaciones internacionales y son exhaustivos y de alto costo en su aplicación.

Hoy en día debido al auge por estudiar carreras relacionadas con las Tecnologías de Información el número de microempresas dedicadas al desarrollo de Software crece cada vez más en nuestro país, para ellas es necesario obtener ganancias en

un tiempo razonablemente corto, de otro modo no podrían subsistir, entonces deben buscar las soluciones a través del desarrollo de proyectos que les permitan obtener los ingresos económicos para mantenerse en el mercado. Al realizar estos proyectos el número de personal es reducido, debido al tamaño de la microempresa se tienen pocos especialistas en las diversas áreas, por lo tanto no se tiene la certeza de que se apeguen a un estándar para crear sus productos.

Considerando los argumentos anteriores resulta conveniente contar con una guía de desarrollo de proyectos para las microempresas basada en un modelo ya existente como lo es MoproSoft, adoptando sus características principales como son el bajo costo de adopción, facilidad de aplicación y de entendimiento.

## **1.2 Antecedentes del problema**

En esta sección se describen los orígenes del problema respecto a que las microempresas consigan calidad en el desarrollo de Software.

Cuando se logra la calidad en el desarrollo del software el usuario espera que se cumplan (Pressman, 2005):

- Los requisitos de funcionalidad y desempeño explícitamente establecidos.
- Los estándares de desarrollo implícitamente documentados.
- Las características implícitas que generalmente se esperan de todo Software desarrollado profesionalmente.

Estas características esperadas están en función de lo que el usuario utiliza en su área de trabajo o uso que hace del Software y cada vez es mayor la diversidad que exige el usuario, por lo que las empresas buscan la mejora continua.

“Descrito en forma general, una empresa está en la búsqueda de una solución inteligente al planteamiento de un problema a resolver, entre muchas, una necesidad humana” (Goodman, 1988). Las empresas desarrolladoras de Software deben apegarse a estándares de calidad ya probados para poder garantizarles un buen producto a sus clientes, pero son difíciles de adaptar por parte de las microempresas, y por ello es necesario que adapten un estándar que puedan cubrir eficientemente y les sea fácil seguir.

Un modelo de procesos debe contener información acerca del proceso de negocio, de la meta que se persigue con la ejecución del proceso, de los problemas que impiden el logro de la meta, entre otra información (Ericsson y Pender, 2000). Sin él no se lograrán los objetivos que se plantearon al inicio del proyecto, en caso de lograrlos se empleará un tiempo mayor al deseado.

### **1.3 Síntesis de trabajos relacionados**

En esta sección se mencionan trabajos de diversos autores relacionados en cuanto a la adopción de modelos de calidad, MoproSoft y algunas situaciones laborales de las microempresas desarrolladoras de Software.

Se pueden corresponder los elementos de los flujos de trabajo con los elementos de MoproSoft (Valenzuela y Flores, 2008), se puede presentar una correspondencia entre los elementos de MoproSoft con los elementos de referencia de flujos de trabajo y el meta-modelo para la definición de procesos, así como la posibilidad de especificar formalmente MoproSoft. Esto brindaría a algunas empresas la posibilidad de aplicar tecnologías de flujo de trabajo para la implantación de MoproSoft. En dicho documento exponen la iniciativa que permitiría a las MiPyMEs modificar gráficamente el flujo de los procesos definidos por MoproSoft y ajustarlo a sus necesidades, pero no se enfocan en la parte de la problemática de las microempresas para adaptar dicho modelo (Flores y Mendoza, 2011).

Existen propuestas de una reestructuración de los diagramas de actividad de acuerdo al nivel de madurez de los procesos (Valenzuela, Flores y Olguín, 2006). Describen como al existir un modelo de procesos para la industria mexicana de software surgió la necesidad de contar con herramientas para su implantación. Flores y Mendoza exponen la metodología utilizada para la interpretación e implantación de MoproSoft por niveles de capacidad de procesos y el modelo de requisitos de una arquitectura base para lograrlo (Flores y Mendoza, 2011).

Es importante analizar las dificultades que tienen las PyMES para adaptar MoproSoft y la necesidad de éstas de hacerlo, así como considerar que algunos miembros del equipo de desarrollo desempeñan más de un rol lo que evidencia la falta de capacitación y de prácticas de Ingeniería de Software (Astorga, Olguín y Flores, 2006). No mencionan cómo el compromiso de la dirección toma otra forma al ser una microempresa.

Existe la posibilidad de evaluar el nivel de adaptación que se tiene de MoproSoft en las actividades del Proceso de Administración de Proyectos Específicos y proponer como trabajo futuro el adaptar, mejorar y agregar a MoproSoft a un Modelo de Madurez (Mon, Estayno y Arancio, 2008).

Existen trabajos que describen una propuesta en la que cualquier empresario con conocimiento mínimo de administración puede lograr un buen producto de Software (Luna, Chaparro, Rojas y Romero, 2004), pero no se apegan a un estándar de calidad ya probado como es MoproSoft.

En el año 2007, se realizó una encuesta y se obtuvo información de parte de 114 empresas dedicadas al desarrollo de software. Los resultados obtenidos indican que “las empresas tienen un conocimiento reducido de los modelos de calidad de software, desconocen los métodos específicos para evaluar la calidad de su producto

y, finalmente, expresan la necesidad de contar con modelos integrales que valoren la calidad del proceso y del producto”. Algunos de los resultados de la encuesta fueron (Gutiérrez, Gutiérrez, Pérez y Márquez, 2009):

El número de personas involucradas en la elaboración del producto software:

- En un 36% de las empresas encuestadas existen de 3 a 5 personas participantes.
- Un 28% tienen de 6 a 10 personas.
- El 21% con 21 o más personas.
- Finalmente, un 15% que tiene entre 11 y 20 personas.

Por lo tanto, se deduce que más de un tercio de las empresas encuestadas (36%) cuentan con menos de cinco personas para la realización del producto software.

Definiendo cuál proceso o metodología utilizan, se establece, que:

- El primer lugar lo ocupan las “metodologías propias” con un 45.41%.
- El segundo lugar lo ocupan las “metodologías ágiles” con un 41.25%; de las cuales las más mencionadas son: XP, Scrum y metodología en espiral.
- En tercer lugar, con 13.34%, se encuentran los “modelos y normas establecidas” como: CMM, CMMI, ISO 9000:2000 y PMBOK.

El 86% de las empresas encuestadas ha considerado utilizar un modelo de aseguramiento de calidad de software. De este porcentaje, en la Figura 1 se muestra que el 44.11% elige a MoproSoft, mientras que el 26.47% se inclina por CMM/CMMI. Por otra parte, el 17.64% no refiere modelos.

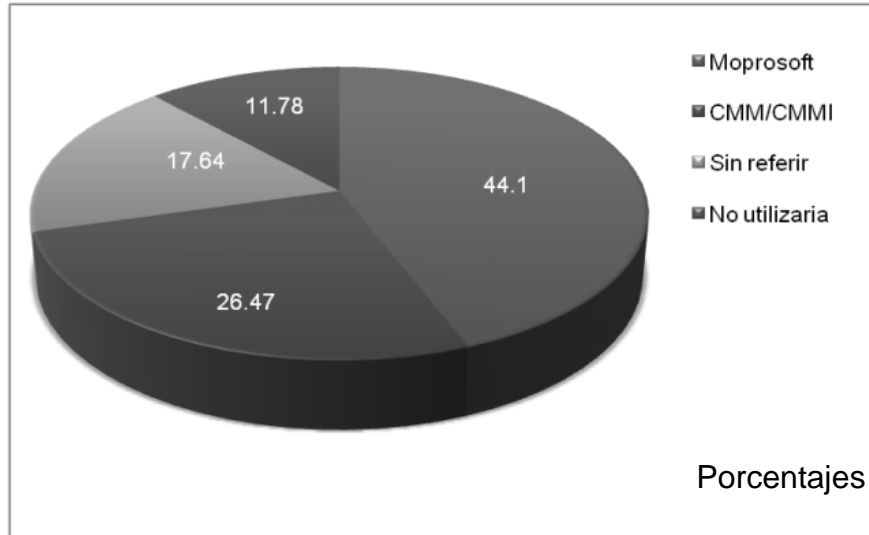


Figura 1 Modelos de calidad de software utilizados

#### 1.4 Planteamiento del problema

En esta sección se mencionan situaciones problemáticas de las microempresas en cuanto a la adopción de modelos de calidad así como porcentajes de las que cumplen con esas características.

“La administración del proyecto de Software permite asegurar que éste se lleve a cabo a tiempo y de acuerdo a la planeación y a los requerimientos” (Mejía, 2004).

México ya dio un gran paso al contar con la norma mexicana NMX-I-059-02 (MoproSoft) dirigida a las PyMES. Hasta el año 2005 sólo se podían encontrar documentadas las experiencias de los casos de estudio del programa de Pruebas controladas, en el que se implantó MoproSoft en cuatro empresas y se evaluaron sus procesos utilizando el Método de Evaluación de Procesos de Software (EvalProSoft) (Alquicira y Su, 2005).

Una de las características más generales de las Micro, Pequeñas y, en algunos casos, de las Medianas empresas, es el uso intensivo de mano de obra no calificada. Esto origina que las remuneraciones percibidas por los trabajadores sean bajas, lo que provoca entre otras cosas una rotación alta de personal. Hay que sumar que en muchas ocasiones los trabajadores no perciben todas las prestaciones que señala la ley (Espinosa y Pérez, 2007). Una microempresa es aquella que está compuesta por menos de 10 personas (Dopacio, 2004).

El proceso de Software se está convirtiendo en una preocupación mayor para las empresas desarrolladoras de Software como una de las maneras de asegurar la calidad mientras se desarrolla un sistema con el proceso de Software.

## **1.5 Hipótesis**

MoproSoft puede ser reducido y adecuado a las microempresas, creando un modelo de calidad para ellas basándose en las prácticas más comunes y obtener un modelo sobre el cual se pueda desarrollar un sistema.

## **1.6 Objetivo General**

Establecer un modelo de calidad para microempresas desarrolladoras de Software que agilice los procesos y reduzca el tiempo de entrega del producto final.

### **1.6.2 Objetivos Específicos**

- Identificar las características de MoproSoft que son viables en un ambiente de desarrollo de una microempresa.
- Conocer los resultados de la implementación de MoproSoft en empresas de mayor tamaño.



- Obtener información sobre la satisfacción de los usuarios de MoproSoft y las ideas que puedan aportar para el desarrollo del nuevo modelo.
- Buscar escenarios propios para la experimentación del nuevo modelo a desarrollar.

## 1.7 Metodologías Empleadas

La metodología es híbrida debido a los diversos métodos empleados y se basa en el Proceso Racional Unificado llamada RUP por siglas en inglés.

RUP es una metodología de desarrollo para proyectos de Software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Son características esenciales (Gómez, 2007):

- Se dirige por los casos de uso: que orientan el proyecto, a la importancia para el usuario y lo que éste quiere
- Está centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, es iterativo e incremental: divide el proyecto en miniproyectos para cumplir objetivos de manera más depurada.

A su vez divide en 4 fases el desarrollo del Software (Mendoza, 2004):

- **Inicio:** El objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** Etapa en la que el objetivo es obtener la capacidad operacional inicial.
- **Transmisión:** Su objetivo es obtener el entregable del proyecto

Para cumplir con la metodología primeramente se realizó una búsqueda de libros, artículos, revistas, ponencias, etc., en gran parte con el apoyo de los asesores en cuanto a la recopilación de artículos científicos, lo que sirvió para definir un estado del arte e identificar lo que se ha hecho en el área en lo referente a MoproSoft y la adopción de estándares de calidad por parte de las microempresas.

Dentro de la definición del estado del arte se realizó una investigación de Modelos de Calidad y las causas que motivaron el origen de MoproSoft. Posteriormente, se revisaron los trabajos relacionados o similares en adopción de Modelos de Calidad, después se buscaron los detalles de una microempresa tales como el número de personal y el comportamiento laboral que tienen dentro de ésta y el modo utilizado para implementar otros modelos en las empresas que desarrollan Software. De igual modo se buscaron las situaciones que dificultan la adopción de Modelos de Calidad, es por eso que se propuso reducir en lo posible MoproSoft para hacer más fácil su adopción. Para identificar los procesos y áreas e inclusive el modo más viable para adaptar el nuevo modelo se tuvo el apoyo de personas que ya trabajaron y cuentan con certificación de MoproSoft.

En lo que respecta al modelo de calidad para la microempresa basado en MoproSoft se buscaron los tipos de Metodologías, resultando las de tipo ágil como las más apegadas a la forma de trabajo de las microempresas de acuerdo a sus características. Posteriormente se buscaron artículos donde se presentaran resultados de las metodologías más utilizadas en las microempresas del país, dicha información concuerda con lo descrito en este trabajo en cuanto a que las microempresas prefieren adoptar Metodologías ágiles.

Dentro de los artículos sobre microempresas se encontró que MoproSoft es utilizado en gran parte de ellas y también cuentan con menos de 10 personas laborando.

Con base en MoproSoft se realizó una comparativa, por lo que se descartaron los procesos a nivel Gerencia que son muy pesados o difíciles de cubrir para las microempresas de acuerdo a sus capacidades y se eliminó el nivel de Alta dirección, asumiendo que en una microempresa este nivel es cubierto por el gerente o dueño.

Respecto al nivel de Operación se realizó una propuesta nueva basándose en los resultados que indican que las Metodologías ágiles más utilizadas son la Extreme Programming (Xp) y Scrum, lo que originó una unión de estas metodologías, dicha combinación ya ha sido propuesta y probada por diversos autores.

Posteriormente, se describieron los procesos del nuevo modelo respetando tal y cual como están descritos por MoproSoft los requisitos de procesos, se ejemplificó esta propuesta con un trabajo realizado en la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad de Colima. Aunque este trabajo no fue realizado por una microempresa se describió como tal con base a que fue un proyecto con el mismo número de personal involucrado y de acuerdo a artículos científicos se realizó en el mismo tiempo que le tomaría a una microempresa.

Por último se hizo una discusión de los elementos del modelo y se demostró el cumplimiento de la hipótesis de trabajo, así mismo se hacen recomendaciones de trabajo a futuro tomando en cuenta lo descrito en este trabajo y el de los autores interesados en adoptar modelos de calidad en las microempresas.

## **1.8 Organización del documento**

El presente documento se encuentra organizado en Introducción y en:

- **Estado del Arte:** En esta sección se describen algunos Modelos de calidad utilizados en la industria del desarrollo de Software y se detallan los niveles de organización con los que cuenta MoproSoft.

- **Modelo de calidad para la microempresa basado en MoproSoft:** En esta sección se detallan los procesos y procedimientos con que cuenta el modelo de calidad que resulta del trabajo de Tesis.
- **Especificación de procesos conforme a MoproSoft:** En esta sección se muestran diversos ejemplos para especificar los procesos del nuevo modelo conforme a MoproSoft.
- **Pruebas y resultados:** En esta sección se describen los resultados de las pruebas aplicadas al modelo de calidad y comparándolo con modelos ya existentes.
- **Conclusiones:** Se hace una síntesis de los resultados obtenidos del trabajo y se proponen trabajos a futuro.

## CAPÍTULO II. ESTADO DEL ARTE

### 2.1 Marco Histórico

En esta sección se describen algunos modelos de calidad utilizados en la industria del desarrollo de Software y se detallan los niveles de organización con los que cuenta MoproSoft.

Uno de los modelos de calidad es ISO 12207, el cual establece un marco de referencia común para los procesos del ciclo de vida Software con una terminología bien definida para utilizarse por la industria del Software. En este marco se definen los procesos, actividades y tareas presentes en la adquisición, suministro, desarrollo, operación y mantenimiento del Software. Las actividades de ISO 12207 se agrupan en (Ruiz y Polo, 2008):

- **Procesos principales:** adquisición, suministro, desarrollo, explotación y mantenimiento.
- **Procesos de soporte:** documentación, gestión de configuración, aseguramiento de calidad, verificación, validación, revisión conjunta, auditoría, resolución de problemas.
- **Procesos organizativos:** gestión, mejora, infraestructura, formación.

También existe el Modelo de Capacidad de Madurez, mejor conocido como CMM, el cual presenta los procesos de desarrollo del Software en una escala de cinco niveles (Vega, Rivera y García, 2008):

- **Caos:** en la empresa no existe ningún modelo y todo se hace sobre la marcha.
- **Repetible:** en este nivel están las empresas en las que existe planificación y seguimiento de proyectos y está implementada la gestión de los mismos.

- **Definido:** en este nivel se documentan y normalizan los procesos a nivel organizativo.
- **Medible:** se pone énfasis en la calidad del proceso y del producto.
- **Mejora continua:** Las áreas clave del proceso incluyen prevención de defectos, administración de cambios tecnológicos y gestión de cambios en los procesos.

MoproSoft identifica los procesos empleados por las empresas de desarrollo y mantenimiento de Software y los agrupa en tres categorías (Oktaba, 2005):

- **Categoría de alta dirección:** Es la que establece la razón de ser de la organización, objetivos y condiciones para lograrlos, así como evaluar los resultados para proponer cambios en busca de una mejora continua.
- **Categoría de gerencia:** Se encarga de las prácticas de gestión de procesos de acuerdo a lo requerido en el plan estratégico, de los proyectos asegurando que cumplan los objetivos y se apeguen a las estrategias, y de los recursos en función de los procesos establecidos a través de la alta dirección.
- **Categoría de operación:** Es la que realiza las prácticas de desarrollo y mantenimiento de Software con los requerimientos especificados, apoyándose en el análisis, diseño, construcción, integración y pruebas.

## 2.2 Marco Contextual

En esta sección se describe el contexto de uso de los modelos de calidad en las microempresas así como experiencias de adopción de éstos mismos.

A partir de que la Secretaría de Economía definió, en el año 2001, el Programa para el Desarrollo de la Industria del Software (PROSOFT), las MiPyMES orientadas a la industria de Software se agruparon en Clusters, esto como un medio para elevar

y extender la competitividad del país, mediante la estrategia de promover el uso y aprovechamiento de las Tecnologías de Información (TI) (PROSOFT, 2005).

Se realizaron numerosos casos de estudio sobre los factores claves para que las PyMES pongan en marcha la iniciativa de mejora de procesos de Software (Rainer y Hall, 2003), en ninguno de los casos se definió una metodología precisa para la investigación y en su mayoría se basan en el seguimiento de sus experiencias, dichos estudios coinciden en algunos factores que se consideran de mayor impacto como: involucramiento de la dirección, participación del personal, métricas, capacitación, consultoría, estándares y procedimientos, personal experimentado, procesos internos, comunicación (Guerrero y Etevoric, 2004; Dyba, 2005).

La escasa experiencia en la implantación de modelos de procesos de Software en las PyMES provoca un desconocimiento sobre los factores que influyen en el éxito de la mejora y en cómo el entorno de la propia organización determina los factores que influyen en la implantación de los modelos de procesos (Rainer y Hall, 2003).

Una de las posibilidades de que las PyMES sean competitivas es ser subcontratadas por grandes empresas (Oktaba2, 2005), en muchas ocasiones esta situación las hace dependientes de proyectos de empresas de mayor tamaño.

Las PyMES de la industria del Software poseen un conjunto de características que presentan serias dificultades para adecuar sus actividades a un proceso de certificación, dadas las pocas facilidades de financiación, los problemas para planear su crecimiento, la falta de gerenciamiento profesional, las dificultades para exportar, y los sistemas de información, administración y contabilidad deficientes.

Pero las PyMES poseen una serie de ventajas dado que poseen mayor posibilidad de flexibilidad y de reacción frente a los cambios, mayor poder de innovación,

menores costos de infraestructura, puntos de ventas cercanos al consumidor, atención más directa y personalizada con los clientes.

Para proponer y conseguir un modelo de desarrollo de Software de calidad es clave aprovechar las ventajas de las microempresas como la atención directa que ofrecen, la cual será de gran ayuda en la propuesta del modelo a través del apoyo e ideas que puedan aportar de acuerdo a sus necesidades y desventajas. Es importante llegar a un punto donde el modelo pueda adaptarse a cualquier microempresa y no sólo a unas cuantas.

### **2.3 Marco Teórico**

En esta sección se describen los fundamentos de las metodologías ágiles mismos que serán la base del modelo desarrollado.

Para asegurar el éxito durante el desarrollo de Software no es suficiente contar con notaciones de modelado y herramientas, hace falta un elemento importante: la metodología de desarrollo, la cual provee de una dirección a seguir para la correcta aplicación de los demás elementos (Calderón, Dámaris, Rebaza y Carlos, 2004).

“Generalmente el proceso de desarrollo llevaba asociado un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada. Este esquema “tradicional” para el desarrollo de Software demuestra ser efectivo y necesario en proyectos de gran tamaño (respecto a tiempo y recursos)”. Sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema está sujeto a grandes cambios, y en donde se exige reducir los tiempos de desarrollo conservando una alta calidad. Por estar especialmente orientadas para proyectos pequeños, las Metodologías Ágiles



constituyen una solución a medida para ese entorno (Calderón, Dámaris, Rebaza y Carlos, 2004).

En una reunión celebrada en el año 2001 nació el término "ágil" aplicado al desarrollo de Software. Tras esta reunión se creó *The Agile Alliance*, una organización dedicada a promover los conceptos relacionados con el desarrollo ágil de Software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto ágil, un documento que resume la filosofía "ágil". El Manifiesto comienza enumerando los principales valores del desarrollo ágil (Canós, Letelier y Penadés, 2010):

- El individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto de Software.
- Desarrollar Software que funciona más que conseguir una buena documentación.
- La colaboración con el cliente más que la negociación de un contrato.
- Responder a los cambios más que seguir estrictamente un plan. La habilidad de la empresa de responder a los cambios determina el éxito o el fracaso de la misma.

La forma de trabajo en las microempresas se apega a los puntos del Manifiesto ágil, además de ser de gran relevancia, ya que el impacto de las metodologías ágiles en la calidad fue en mayoría algo mejor o mucho mejor, como se muestra en la Figura 2, según una encuesta en Febrero 2008 donde participaron 642 personas (Ambler, 2008).

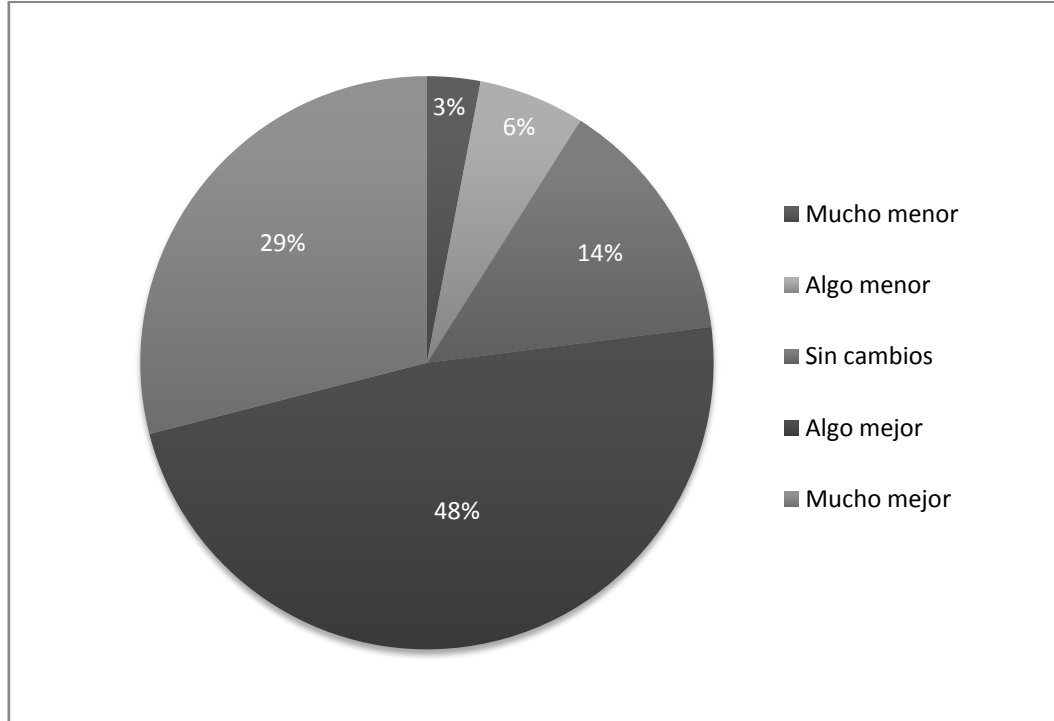


Figura 2 Impacto de Metodologías ágiles en la calidad (Ambler, 2008).

Así mismo en el año 2007, se realizó una encuesta y se obtuvo información de parte de 114 empresas dedicadas al desarrollo de software. Los resultados obtenidos indican que “las empresas tienen un conocimiento reducido de los modelos de calidad de software, desconocen los métodos específicos para evaluar la calidad de su producto y, finalmente, expresan la necesidad de contar con modelos integrales que valoren la calidad del proceso y del producto” (Gutiérrez, Gutiérrez, Pérez y Márquez, 2008).

A continuación se presentan algunos de los resultados de la encuesta:

Como se muestra en la Figura 3, es muy pequeño el número de personas involucradas en la elaboración del producto software: en un 36% de las empresas encuestadas existen de 3 a 5 personas participantes, seguido de un 28% que tienen de 6 a 10, un 21% con 21 o más y, finalmente, un 15% que tiene entre 11 y 20

personas. Por lo tanto, se deduce que más de un tercio de las empresas encuestadas (36%) cuentan con menos de cinco personas para la realización del producto software (Gutiérrez, Gutiérrez, Pérez y Márquez, 2008).

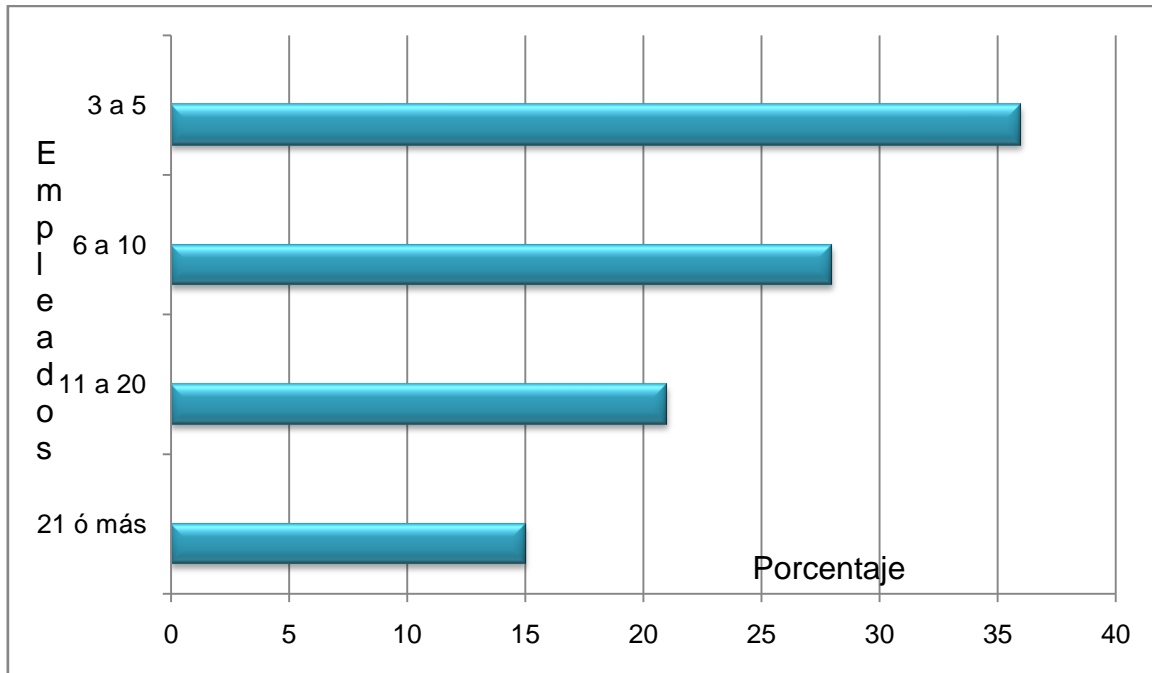


Figura 3 Número de personal en elaboración de Software

Definiendo cuál proceso o metodología utilizan, se establece, Figura 4, que el primer lugar lo ocupan las “metodologías propias” con un 45.41%. El segundo lugar lo ocupan las “metodologías ágiles” con un 41.25%; de estas últimas las más mencionadas son: XP [XP], Scrum [SCRUM] y metodología en espiral. En tercer lugar, con 13.34%, se encuentran los “modelos y normas establecidas” como: CMM [SEI, CMM], CMMI [SEI, CMMI], ISO 9000:2000 [ISO 9000:2000] y PMBOK [PMI, PMBOK] (Gutiérrez, Gutiérrez, Pérez y Márquez, 2008).

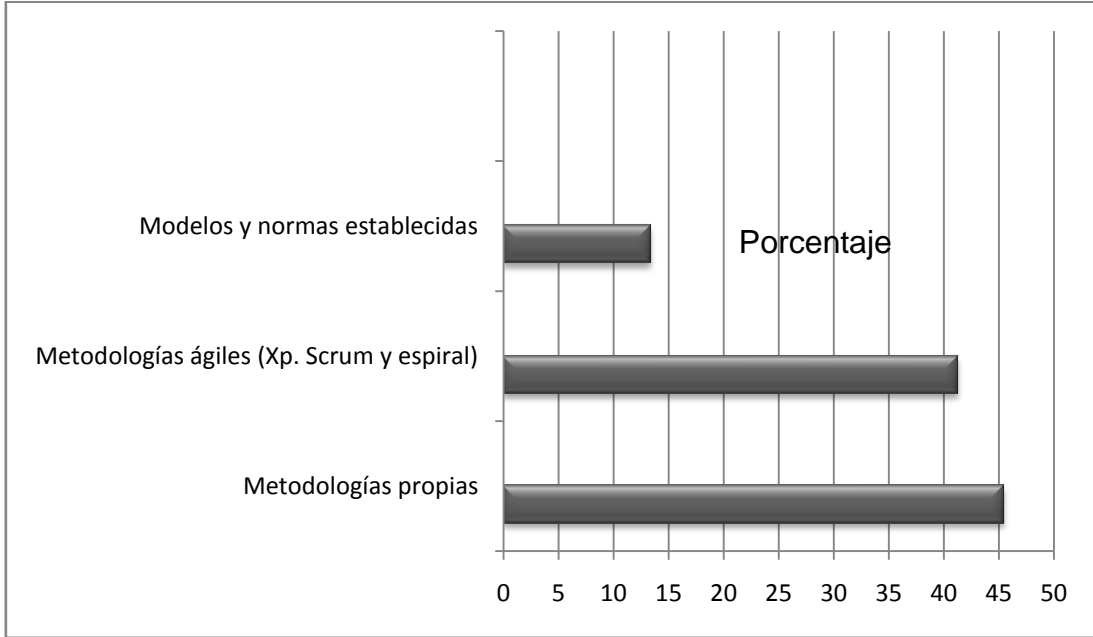


Figura 4 Metodologías utilizadas

Más de la mitad de las empresas que participaron en la encuesta (53%) consideran que la documentación generada por ellos no es de calidad, Figura 5 (Gutiérrez, Gutiérrez, Pérez y Márquez, 2008).

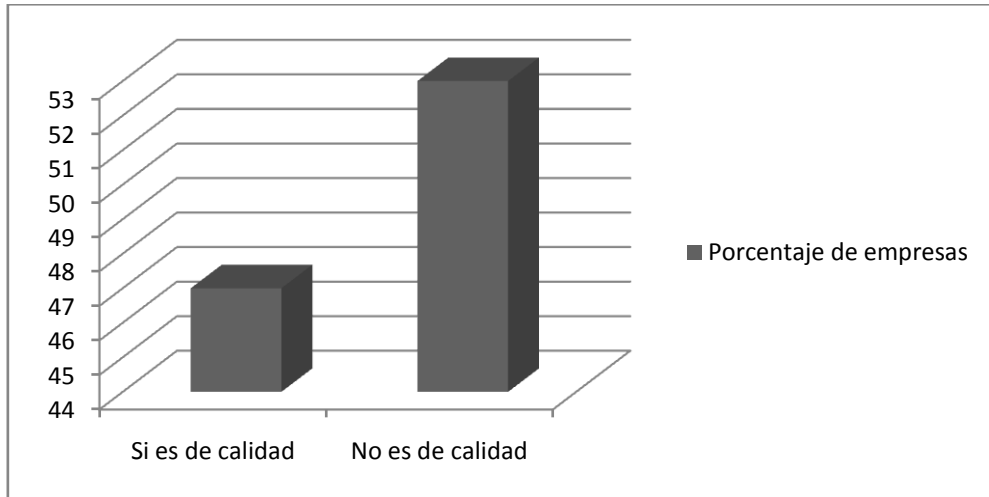


Figura 5 Documentación de calidad

La determinación de que la documentación no es de calidad está referida a que sólo se documenta el manual de usuario por falta de recursos humanos especializados y por considerar a la documentación como elemento accesorio y no como la evidencia de la realización de un proceso, ejecutándose una vez terminado el producto software (Gutiérrez, Gutiérrez, Pérez y Márquez, 2008).

El 86% de las empresas encuestadas ha considerado utilizar un modelo de aseguramiento de calidad de software. De este porcentaje, en la Figura 6 se muestra que el 44.11% elige a MoproSoft, mientras que el 26.47% se inclina por CMM/CMMI. Por otra parte, el 17.64% no refiere modelos (Gutiérrez, Gutiérrez, Pérez y Márquez, 2008).

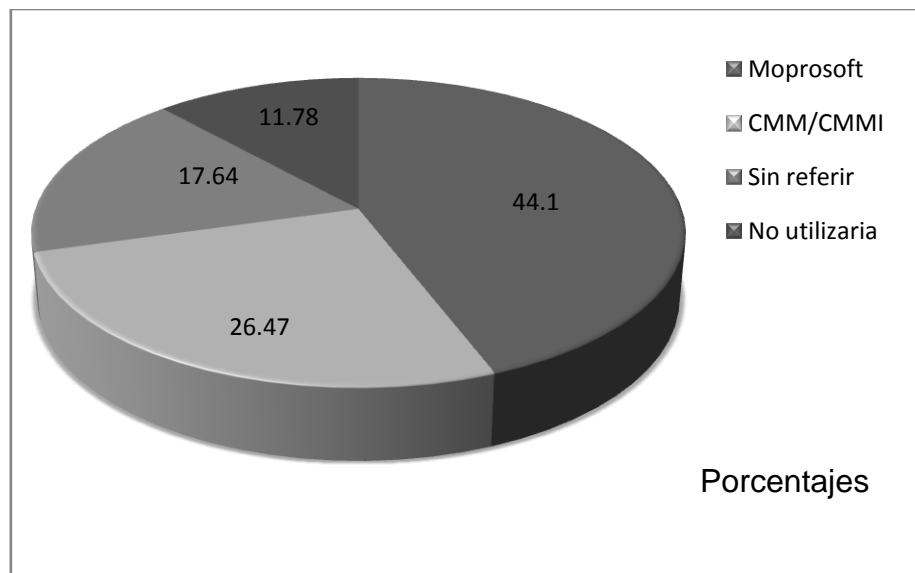


Figura 6 Modelo de calidad de Software

Existe la disposición de las empresas de adoptar un modelo de calidad, generalmente se toma la documentación como un accesorio que se incluye al producto y no como evidencia de lo realizado, dentro de la encuesta se observó que las metodologías más utilizadas son las mismas que servirán de base para el modelo que se propone en éste trabajo.

### CAPITULO III. MODELO DE CALIDAD PARA LA MICROEMPRESA BASADO EN MOPROSOFT

En este capítulo se describe y hace una comparativa del modelo de calidad con MoproSoft. Como se mencionó anteriormente en este trabajo MoproSoft cuenta con tres niveles de organización; Alta dirección, Gerencia y Operación. En el primero de estos niveles se toman las decisiones y se define la misión y valores de la empresa (Figura 7), en una microempresa la alta dirección está conformada por el dueño de la microempresa y él tiene que realizar las labores de Alta dirección y Gerencia al mismo tiempo, por lo que en este trabajo se propone conservar solamente el nivel de Gerencia para la microempresa.

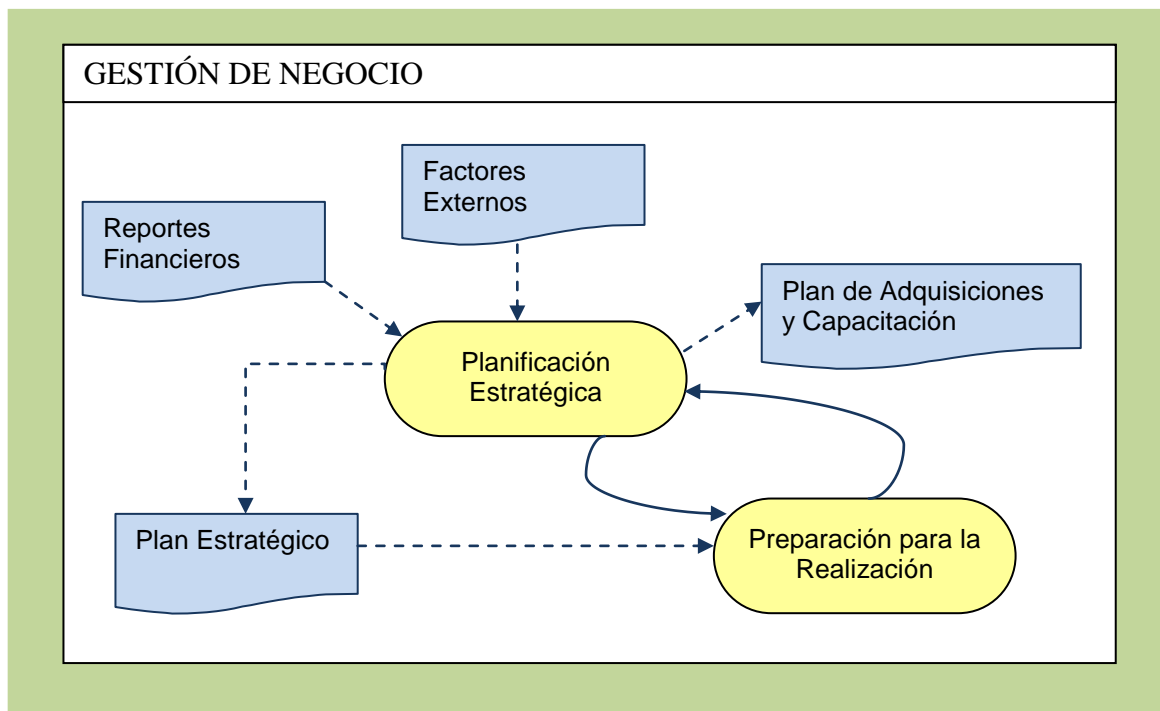


Figura 7 Alta dirección de MoproSoft

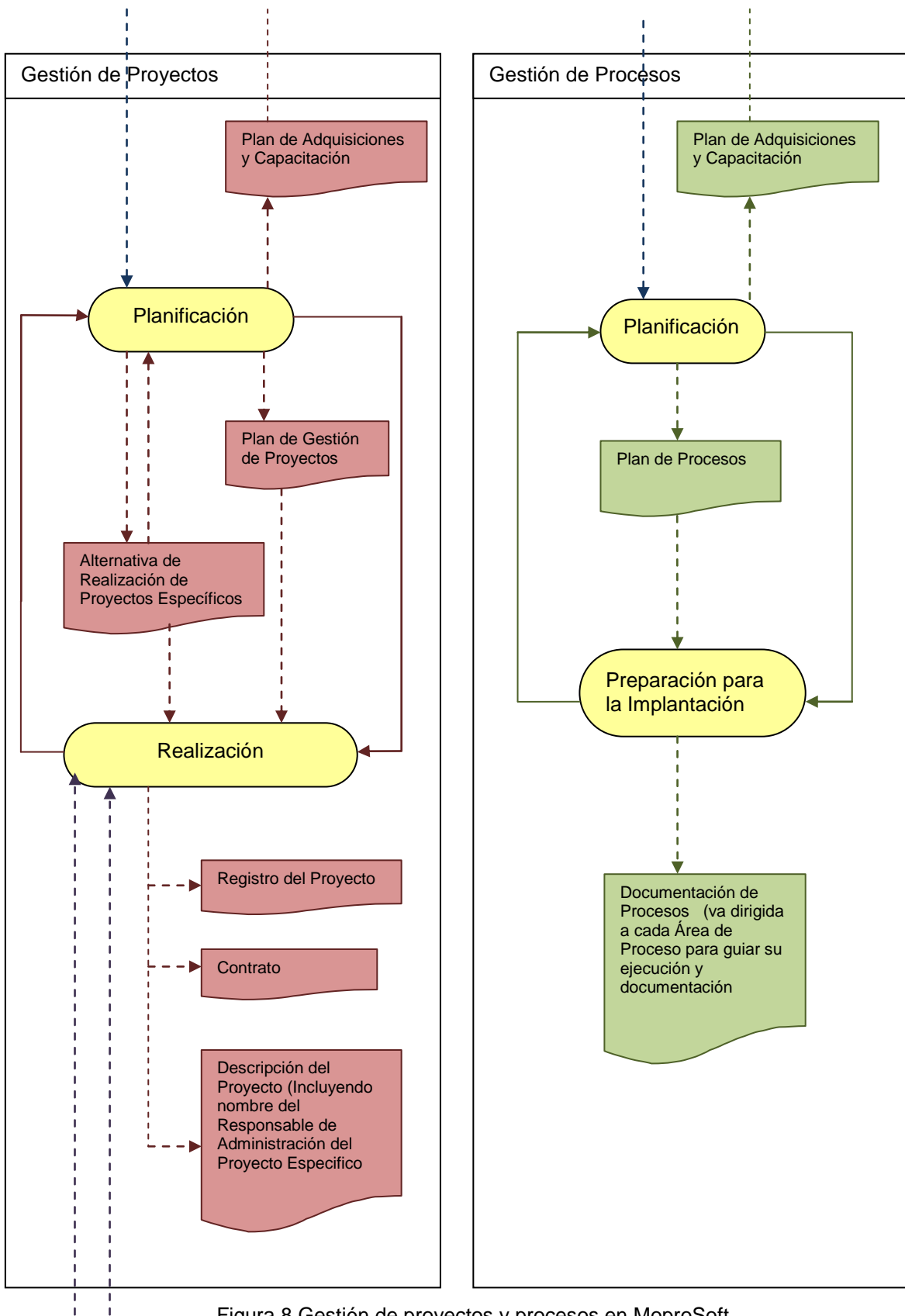


Figura 8 Gestión de proyectos y procesos en MoproSoft

A nivel Gerencia MoproSoft se divide en 3, Gestión de proyectos, Gestión de Procesos (Figura 8) y Gestión de Recursos (Figura 9). Dentro de la Gestión de Procesos se descartan las Alternativas de realización de proyectos específicos ya que como se ha descrito debido al tamaño de la microempresa raramente se pueden enfocar en más de un proyecto al mismo tiempo.

También se descarta el Plan de adquisición y capacitación debido a que en una microempresa generalmente la capacitación es individual, es decir cada individuo se especializa en las áreas que cree necesarias para dar apoyo a la microempresa. Esto debido a los tiempos reducidos para desarrollar productos y a que tienen que estar laborando constantemente para subsistir, lo que deja poco tiempo para la capacitación.

Dentro de la Gestión de Recursos (Figura 9) en el nuevo modelo se descarta el plan de adquisiciones y capacitación por las razones descritas anteriormente. El nuevo modelo conserva el Plan de administración de la base de conocimiento, Diseño de la base de conocimiento y la Base de conocimiento.

En caso de un nuevo proyecto similar a alguno realizado con anterioridad se tendrá en la Base de conocimiento la información necesaria para desarrollar con mayor rapidez el nuevo proyecto debido a las experiencias anteriores. También se adopta el registro de recursos humanos, así se tiene una relación de los empleados y las habilidades y experiencia con la que cuentan.



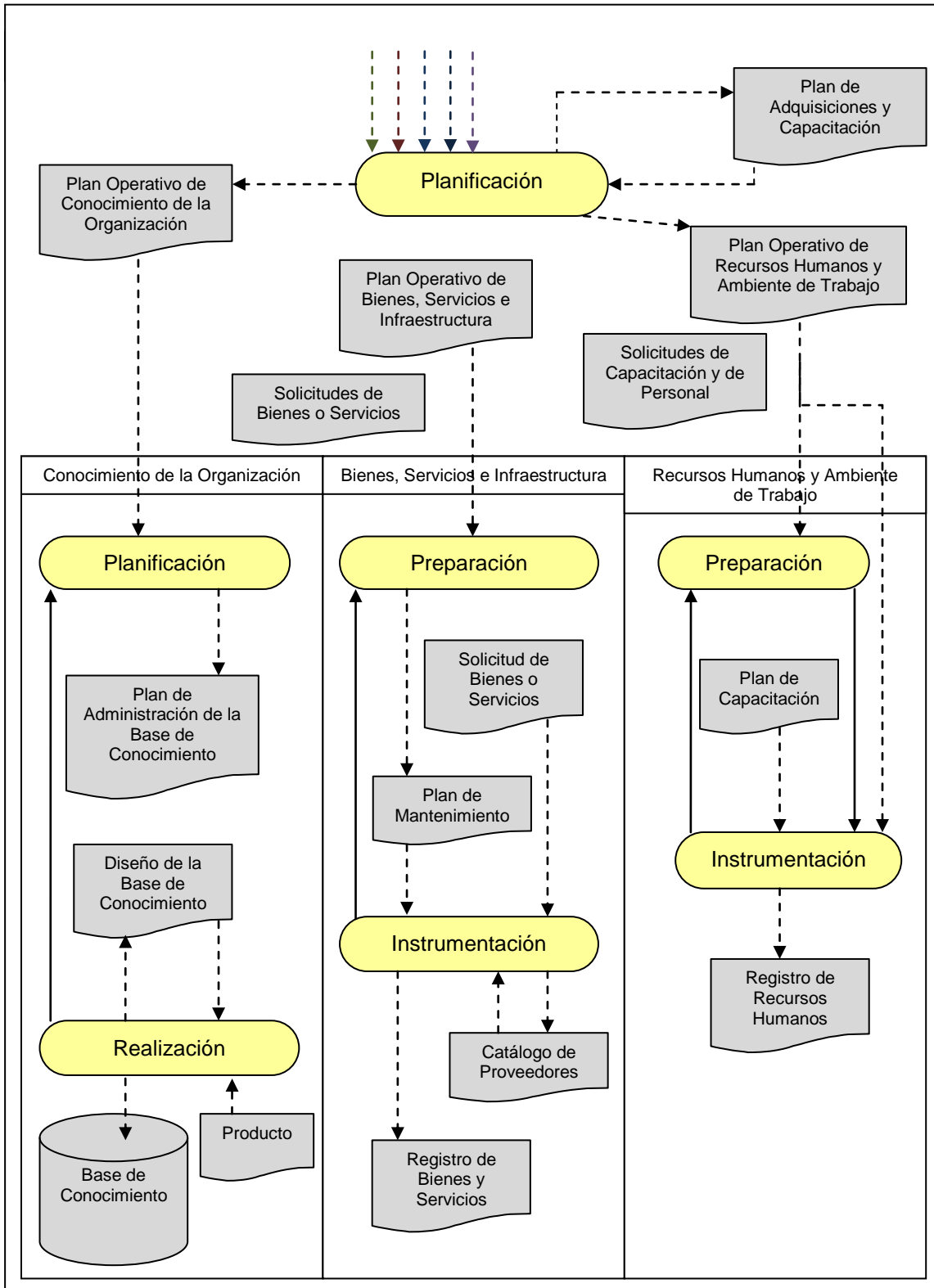


Figura 9 Gestión de recursos en MoproSoft

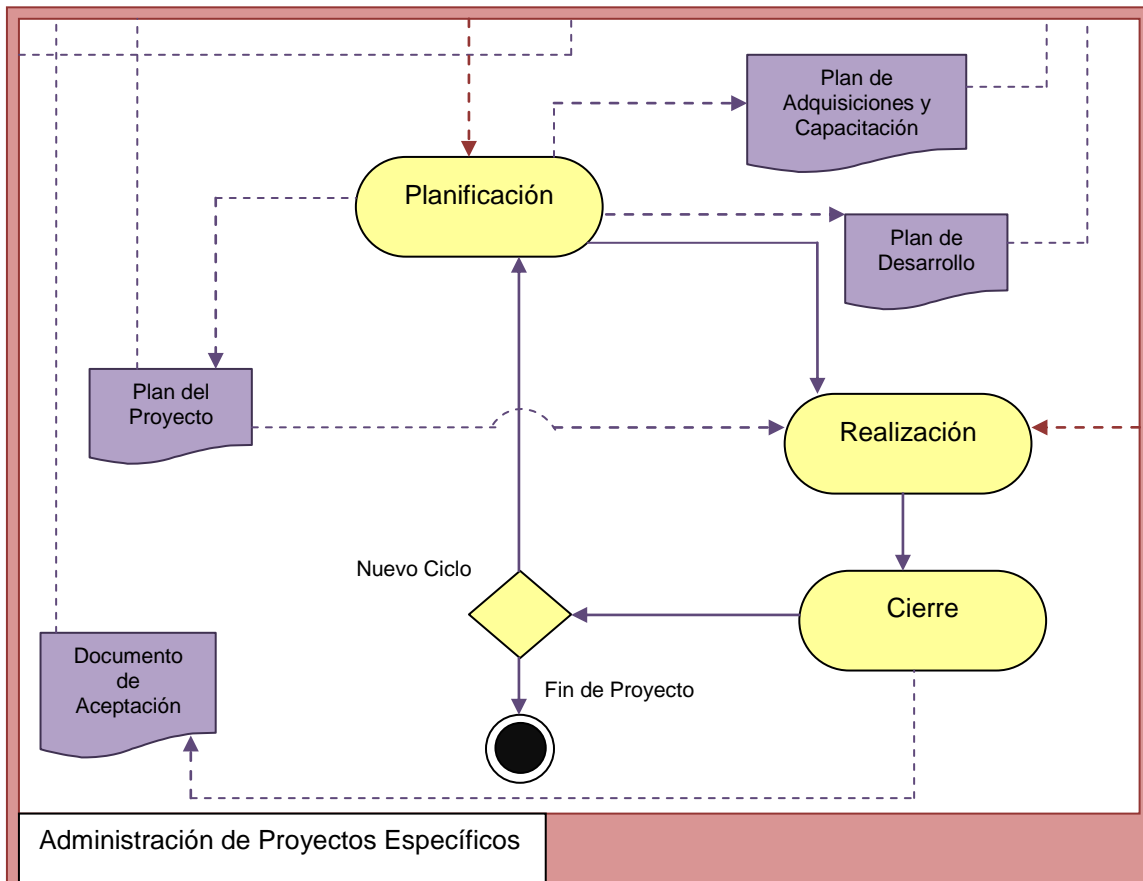


Figura 10 Administración de proyectos específicos en MoproSoft

A nivel Operación MoproSoft cuenta con Administración de Proyectos específicos y Desarrollo y mantenimiento de Software. En el nuevo modelo se conserva la administración de proyectos específicos como lo plantea MoproSoft a excepción del plan de adquisiciones y capacitación (Figura 10). Para el Desarrollo y mantenimiento de Software se propone una metodología diferente a la utilizada por MoproSoft (Figura 11), según los resultados de la encuesta mostrada en el capítulo anterior las metodologías más utilizadas son Extreme Programming (XP) y Scrum, por lo que tratando de abarcar al mayor número de microempresas se propone una unión entre estas metodologías.

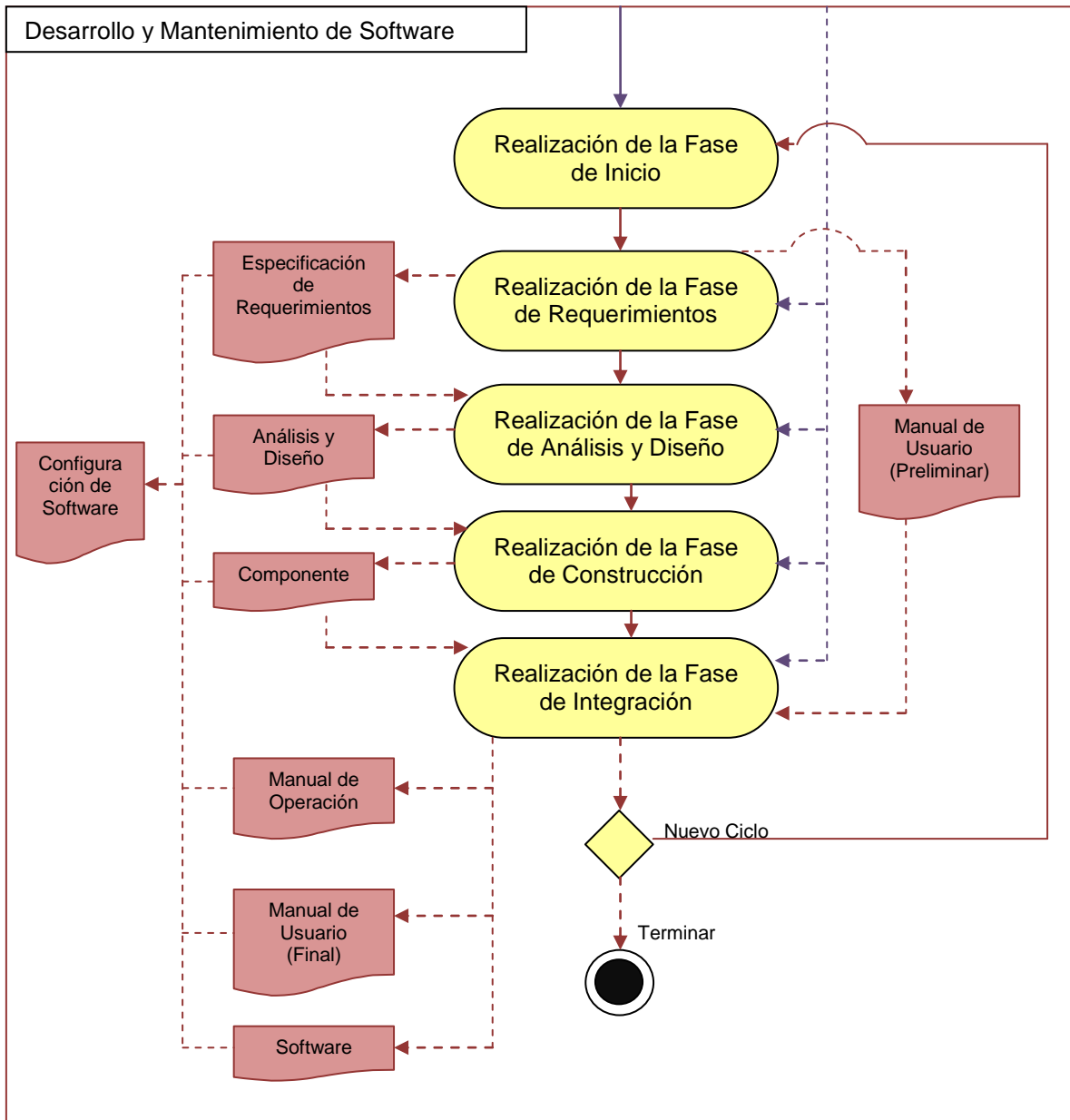


Figura 11 Desarrollo y mantenimiento de Software en MoproSoft

En la propuesta desarrollada el modelo cuenta con 2 niveles organizacionales, Gerencia y Operación (Figura 12). En el nivel de Gerencia se conservan la Gestión de Proyectos, la Gestión de Procesos y la Gestión de Recursos. Dentro de cada una de estas divisiones existen documentos a realizar dentro del desarrollo del proyecto.

En el nivel Operación se tienen 5 fases, Fase de Exploración, Fase de Planificación, Fase de Diseño, Fase de Producción y Fase de Implantación, estas fases están en función de lo planteado en la Gerencia.

En la Tabla 1 se presenta una comparativa entre las principales diferencias de MoproSoft y el modelo propuesto

Tabla 1. Comparativa entre MoproSoft y el modelo propuesto

	MoproSoft	Modelo propuesto
Alta gerencia	Si	Unido a la gerencia
Gerencia	Gestión de proyectos, Gestión de procesos, Gestión de recursos	Gestión de proyectos, Gestión de procesos, Gestión de recursos
Gerencia / gestión de recursos	Plan de capacitación Plan operativo de conocimiento de la organización	Sin plan de capacitación Sin plan operativo de conocimiento de la organización
Operación / administración de proyectos específicos	Si	Modificado, planificación dentro del desarrollo y mantenimiento de software Sin plan de adquisiciones y capacitación
Operación / desarrollo y mantenimiento de software	Fase de inicio, requerimientos, análisis y diseño, construcción e integración Manual de usuario preliminar y final	Fase de exploración, planificación, diseño, construcción, implementación Manual de usuario final

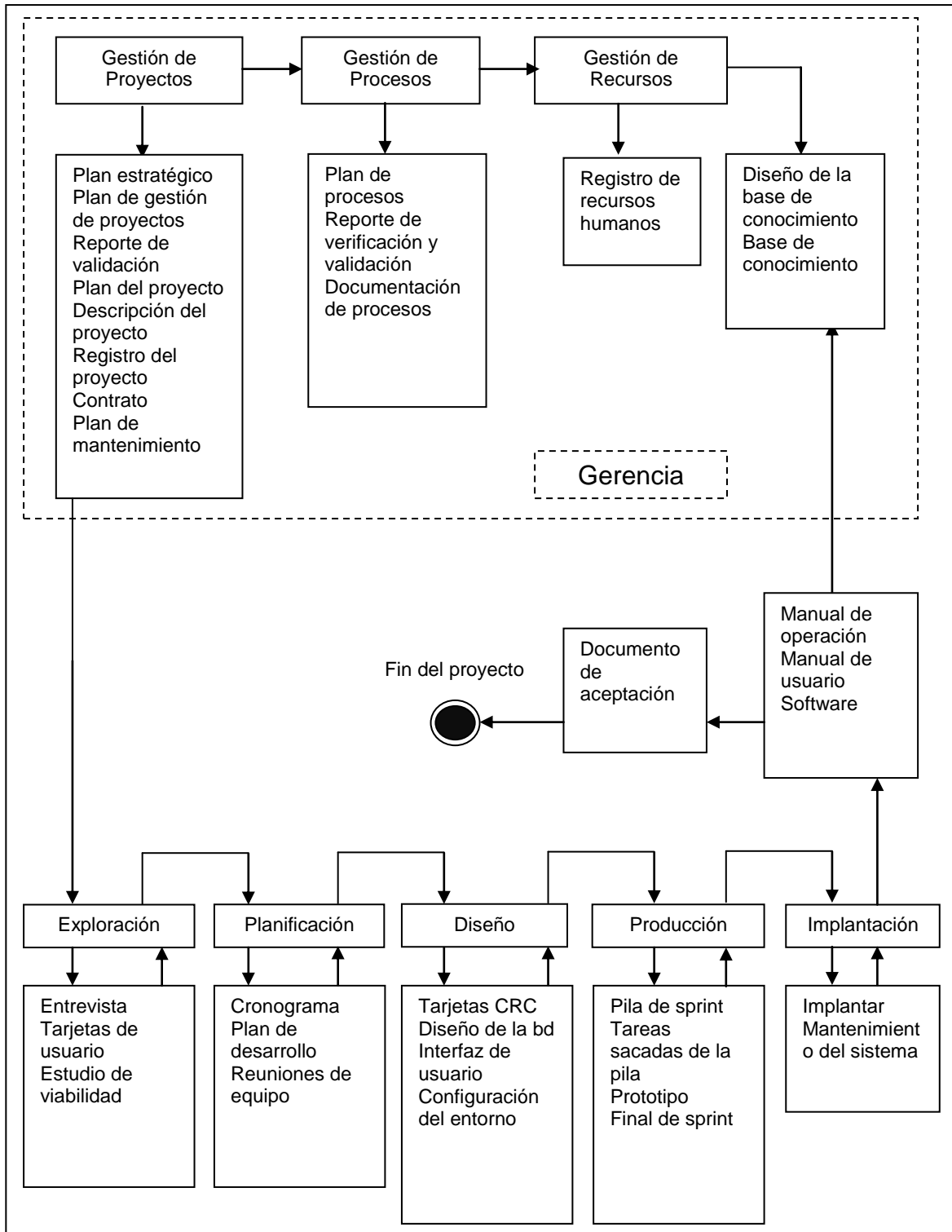


Figura 12 Modelo de calidad para la microempresa basado en MoproSoft



Dentro de la planificación de la Gestión de Proyectos se realizan las siguientes tareas (Norma NMX-I-059/02-NYCE-2005):

- a) A partir del *Plan Estratégico* se establece o actualiza el *Plan de Gestión de Proyecto*, considerando proyectos externos, internos y oportunidades de proyectos de la organización. Este plan contiene dos elementos:
  - i) *Plan de Ventas*, se planea la realización de acciones para generar y cerrar las oportunidades del proyecto, la presentación de propuestas y la firma de *Contrato*;
  - ii) *Plan de Proyecto*, se planea la asignación de recursos, el seguimiento del proyecto interno seleccionado.
- b) Se debe realizar la validación del *Plan de Gestión de Proyecto*, esto generará el *Reporte(s) de Validación*.

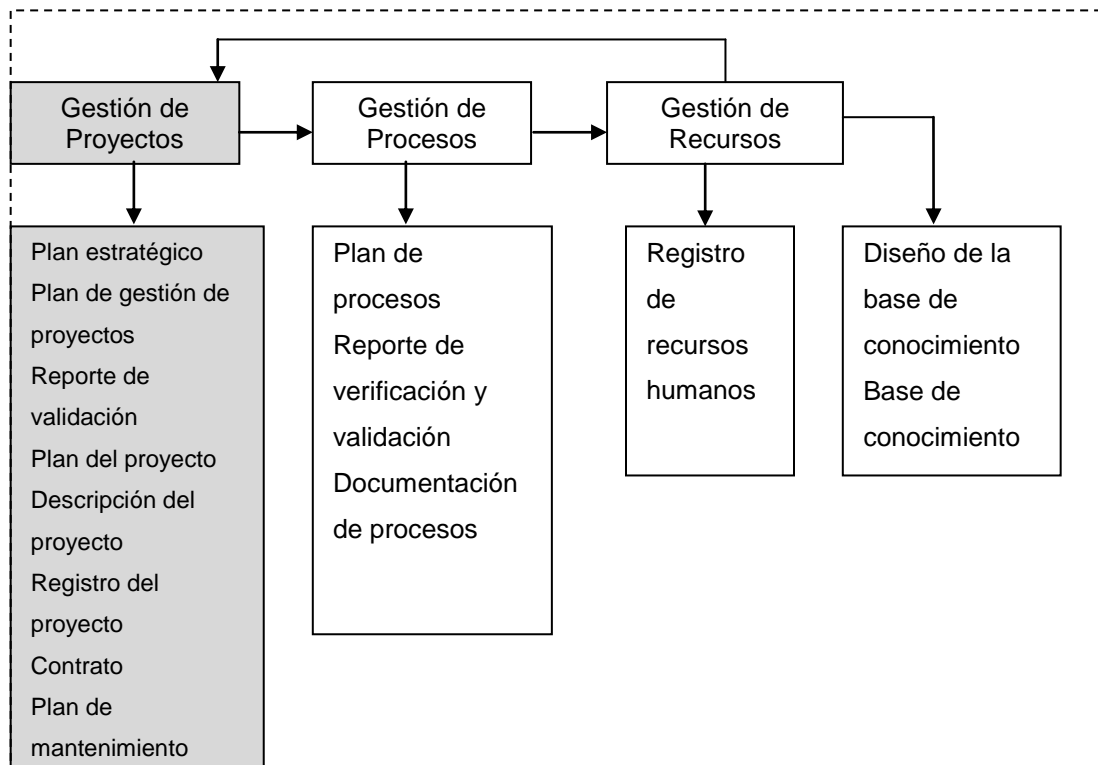


Figura 14 Gestión de Proyectos

En el *Plan Estratégico* se deben establecer los siguientes puntos (Norma NMX-I-059/02-NYCE-2005):

1. Misión: Razón de ser de la organización.
2. Visión: Posición deseada de la organización en el mercado.
3. Valores: Cualidades y virtudes que se comparten entre los miembros de la organización y se desean mantener.
4. Objetivos: Resultados a buscar para cumplir con la Misión y Visión
5. Estrategias: Forma de lograr objetivos
6. Procesos requeridos: Identificación de los procesos con su propósito, objetivos, e indicadores para llevar a cabo las estrategias.
7. Cartera de proyectos: Conjunto de proyectos externos e internos u oportunidades de proyectos.
8. Estructura de la organización: Definición de áreas y responsabilidades de la organización requerida para llevar a cabo las estrategias.
9. Estructura de recursos: Definición, planificación y asignación de recursos en la organización para el cumplimiento de las estrategias, considerando los elementos de la Base de Conocimiento necesarios para el almacenamiento y consulta de la información generada en la organización.
10. Presupuesto: Gastos e ingresos esperados en un periodo determinado.
11. Periodicidad de valoración: Definición de los periodos para realizar las revisiones de valoración y mejora.
12. Plan de Comunicación con el cliente: Definición de los mecanismos para establecer los canales de comunicación con los clientes.

En un *Plan de Ventas* se definen los objetivos, alcances, recursos, acciones y plan de trabajo necesarios para realizar la venta del producto:

1. Objetivos de la Venta
2. Alcances de la Venta
3. Recursos necesarios para realizar la Venta



4. Acciones de Venta
5. Plan de trabajo de Venta

El *Plan de Proyecto* servirá para estimar tiempos y costos, entre otros del desarrollo del producto, dentro de la Operación es nombrado como el Plan de Desarrollo, cuenta con los siguientes puntos:

1. Ciclos y actividades
2. Tiempo estimado
3. Equipo de trabajo
4. Costo estimado
5. Calendario
6. Plan de manejo de riesgos
7. Protocolo de entrega

Al realizar el *Registro del Proyecto* se deben establecer:

1. Nombre del proyecto
2. Responsable
3. Fechas de inicio y terminación
4. Cliente

Además de contar con un *Plan de Mantenimiento* que especificará los responsables de dar mantenimiento al sistema:

- Calendario
- Responsables
- Proveedores

Un elemento indispensable en el proyecto es el *Contrato*, el éxito en los proyectos de desarrollo de software a la medida depende, en gran medida, de que el trabajo entre el cliente y el desarrollador se encuentre debidamente normado. En este

sentido, es un factor reconocido de éxito para estos proyectos formalizar los acuerdos entre ambos, las especificaciones del trabajo por desarrollar, precios, términos y condiciones, entre otros elementos. Las principales ventajas de cuidar la elaboración de un buen contrato informático para el desarrollo de proyectos de software son (Peñaloza, 2006):

- Describe técnica y legalmente las actividades por desarrollar, de manera que un buen contrato debe evitar que dichas actividades queden sujetas a la interpretación.
- Un buen contrato debe incluir relaciones completas y claras de los requerimientos y especificaciones del software que se construirá.
- Documenta los acuerdos verbales tomados por las partes, dichos acuerdos y los compromisos asumidos no podrán ser cambiados ni por el tiempo ni ante eventos futuros, a menos que el mismo contrato lo indique.
- Ayuda a proteger tanto al cliente como al desarrollador ante cualquier “desastre” potencial, lo que previene eventos y riesgos desde la negociación inicial y la definición del producto por desarrollar.
- Define la forma en que se dará por terminado este proyecto y, por lo tanto, la relación legal entre el cliente y el desarrollador.
- Promueve la participación, comunicación e involucramiento de ambas partes para el desahogo y seguimiento de actividades durante el desarrollo del proyecto

Es necesario realizar la *Descripción del Proyecto* para informar al cliente de los alcances y entregables del producto, así como las restricciones, el plan tiene los siguientes puntos (Norma NMX-I-059/02-NYCE-2005):

1. Propósito
2. Productos
3. Objetivos
4. Alcance

5. Entregables
6. Necesidad de negocio
7. Supuestos y premisas
8. Restricciones

### 3.1.2 Gestión de Procesos

El propósito de la Gestión de Procesos (Figura 15) es establecer los procesos de la organización, en función de los *Procesos Requeridos* identificados en el *Plan Estratégico*. Así como definir, planificar, e implantar las actividades de mejora en los mismos.

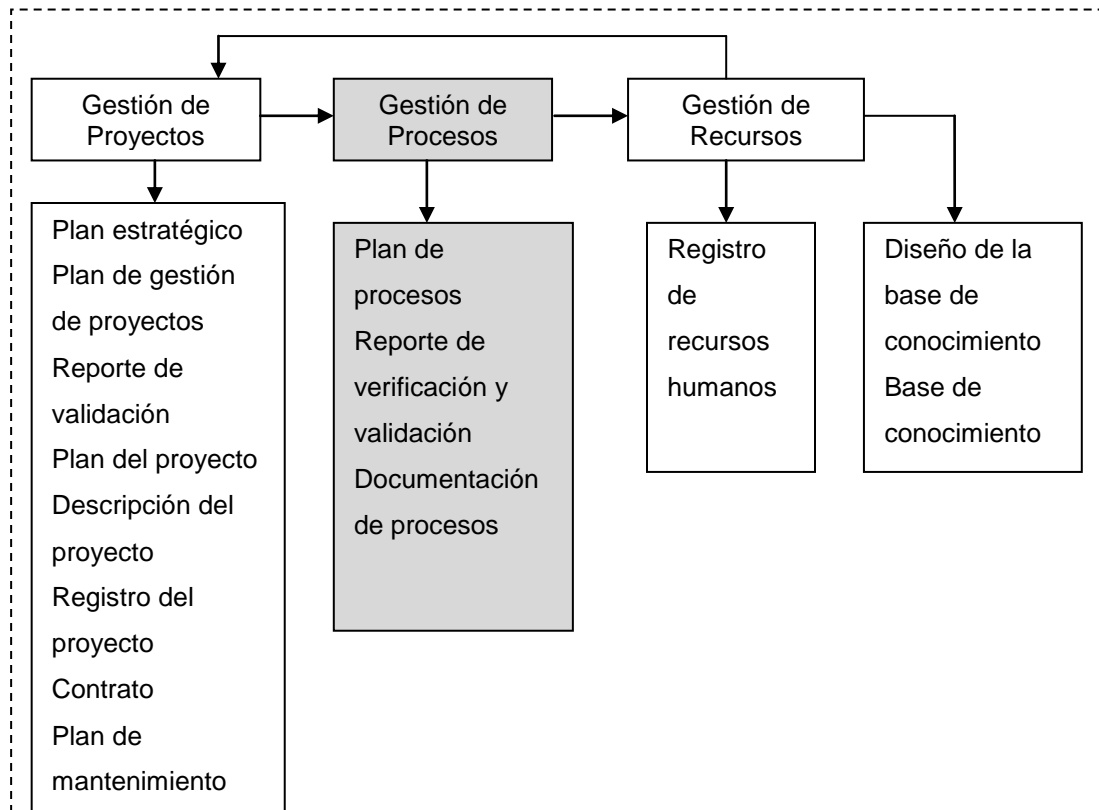


Figura 15 Gestión de Procesos

A partir de los *Procesos Requeridos*, identificados en el *Plan Estratégico*, se establece o se actualiza un *Plan de Procesos*, el cual contiene los siguientes elementos (Norma NMX-I-059/02-NYCE-2005):

1. Definición de elementos de procesos: Contiene los elementos establecidos para los procesos.
2. Calendario: Fechas para actividades en la definición, implantación y mejora de procesos incluyendo responsables.
3. Plan de evaluación: Propone las formas de evaluar procesos. Incluye evaluaciones internas y externas.
4. Plan de manejo de riesgos: Contiene la identificación y evaluación de riesgos, así como los planes de contención y de contingencia correspondientes.

El *Plan de Procesos* se verifica y se valida, esto generara el *Reporte de Verificación y Reporte de Validación*. Dentro del desarrollo del producto se debe realizar una *Documentación de Procesos*, cada uno de ellos con los siguientes puntos (Norma NMX-I-059/02-NYCE-2005):

1. Nombre del proceso
2. Descripción
3. Objetivos
4. Indicadores
5. Metas cuantitativas
6. Responsabilidad y autoridad
7. Procesos relacionados
8. Entradas
9. Salidas
10. Productos internos
11. Roles involucrados

12. Actividades
13. Verificaciones y validaciones
14. Incorporación a la base de conocimiento
15. Recursos de infraestructura
16. Mediciones
17. Situaciones excepcionales
18. Lecciones aprendidas
19. Guías de ajuste.

### **3.1.3 Gestión de Recursos**

Para la Gestión de Recursos (Figura 16) debe existir un *Conocimiento de la Organización*, el propósito es mantener disponible y administrar la *Base de Conocimiento* que contiene la información y los productos generados por la organización. El objetivo es proporcionar a la organización la *Base de Conocimiento* de forma confiable, oportuna y segura mediante el cumplimiento del *Plan de Administración de la Base de Conocimiento*.

En el *Plan de Administración de la Base de Conocimiento* se genera o actualiza la *Base del Conocimiento*, así mismo se valida y se genera un *Reporte de Validación*, también se implantarán o se dará mantenimiento a la *Base de Conocimiento* para poder incorporar y consultar los productos que ya han sido aprobados de entre todos los procesos y proyectos.

Para realizar el *Diseño de la Base de Conocimiento* se deben definir diversos mecanismos, como son (Norma NMX-I-059/02-NYCE-2005):

1. Diseño del modelo conceptual, inclusive metamodelo.
2. Definición de mecanismos de alimentación.
3. Definición de mecanismos de consulta.

4. Definición de mecanismos de control de acceso.
5. Definición de mecanismos de mantenimiento
6. Definición de mecanismos de respaldo.

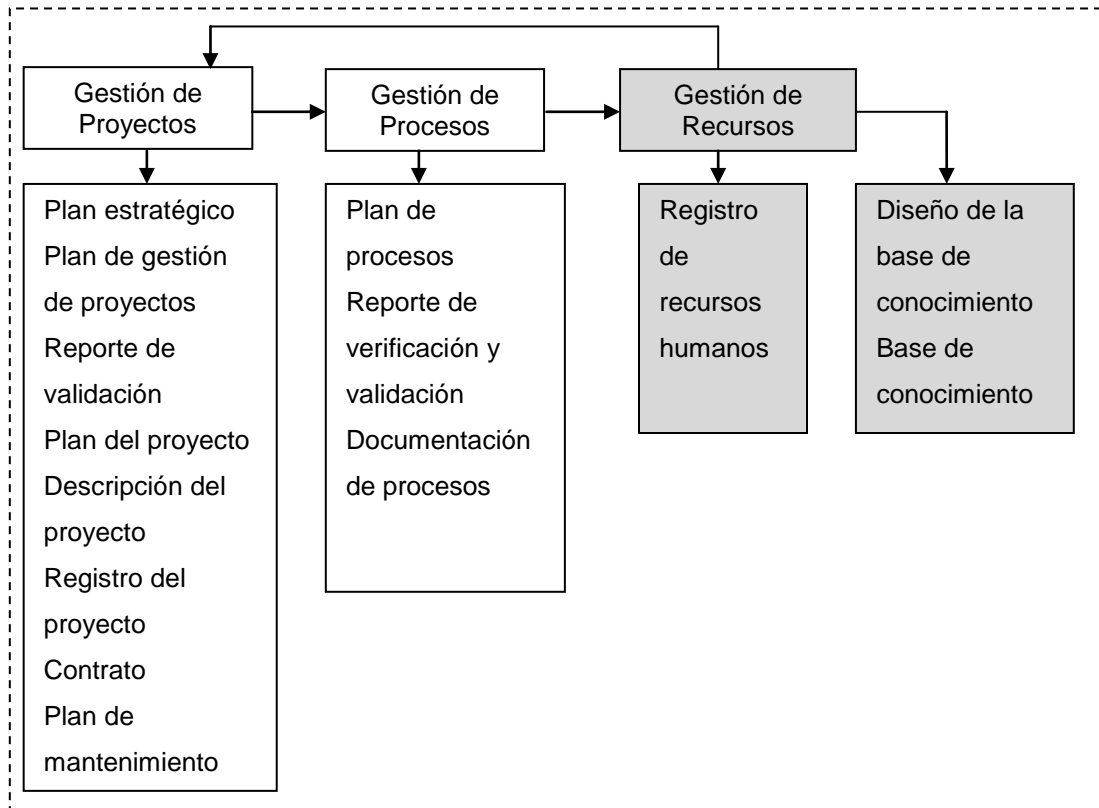


Figura 16 Gestión de Recursos

En la *Base de Conocimiento* se tienen diversos repositorios, producto de lo que ha sido desarrollado por la organización (Norma NMX-I-059/02-NYCE-2005):

1. Repositorio de negocio
2. Repositorio de procesos
3. Repositorio de proyectos
4. Repositorio de desarrollo y mantenimiento
5. Repositorio de recursos
6. Repositorio de recursos humanos

7. Repositorio de bienes adquiridos y proveedores
8. Repositorio de documentación de la base de conocimiento

El producto aprobado deberá incorporarse a la *Base de Conocimiento* de cada área de proceso. También se tiene un Registro de Recursos Humanos, que contiene los siguientes datos (Norma NMX-I-059/02-NYCE-2005):

1. Datos personales
2. Formación
3. Experiencia
4. Roles asignados
5. Evaluaciones de desempeño.

### 3.2 OPERACIÓN

El nivel de Operación (Figura 17) se basa en MoproSoft en su subdivisión de administración de proyectos específicos, pero se propone una nueva metodología de desarrollo en la división de desarrollo y mantenimiento de software, esta propuesta surge de combinar la metodología Extreme Programming (xp), con SCRUM y DSDM

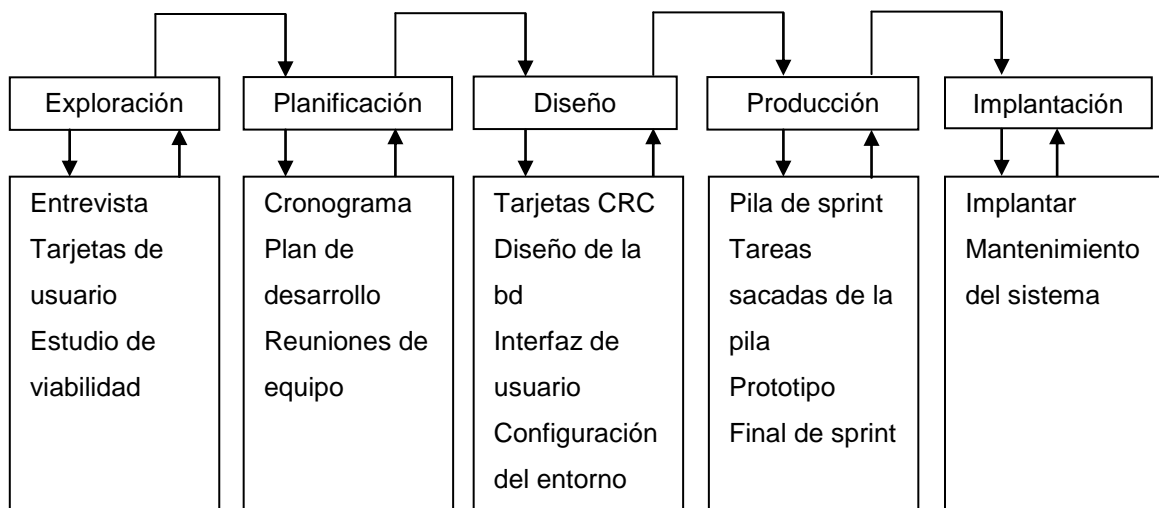


Figura 17 Nivel de Operación

### 3.2.1 Fase de Exploración

En esta fase de exploración se realizan las entrevistas con el usuario para conocer lo que el cliente requiere, posteriormente se realiza un estudio de viabilidad y estudio del negocio según DSDM, siglas en inglés de Dynamic Systems Development Method, la cual define el marco para desarrollar un proceso de producción de software. Nació en 1994 con el objetivo de crear una metodología RAD unificada. Las principales características con las que cuenta: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio de viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación (DSDM, 1994).

En la Fase de exploración (Figura 18) los clientes ponen por escrito en las tarjetas de historia de usuario lo que desean sea incluido en cada entrega. Cada tarjeta de historia describe una característica que se agregará en el programa. Al mismo tiempo el equipo del proyecto se familiariza con las herramientas, la tecnología y las prácticas que utilizarán en el proyecto (Paulk, 2001).

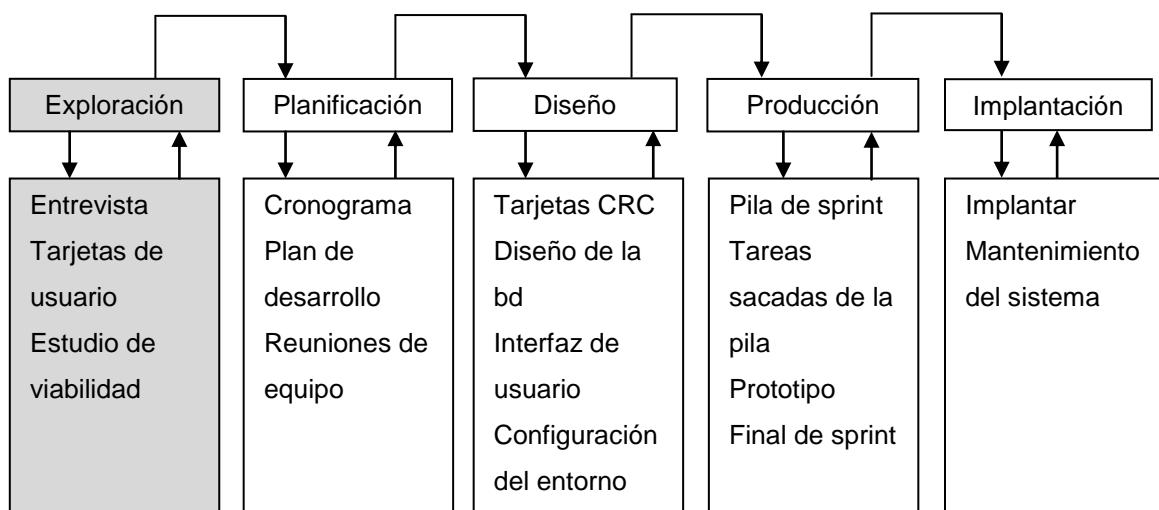


Figura 18 Fase de Exploración



## **Entrevista**

Entre las técnicas utilizadas por los ingenieros de requisitos está la entrevista, que elaboran en conjunto el ingeniero de requisitos y los interesados. Para elaborar una entrevista es necesario, primero, identificar los posibles candidatos a entrevistar. Se debe procurar entrevistar a personas interesadas en el desarrollo del sistema y posibles usuarios finales. Después de identificar los candidatos, se prepara la entrevista, precisando las preguntas a utilizar, el lugar donde se realizará la entrevista, las herramientas que se utilizarán, etc. Finalmente, cuando la entrevista culmina, se extraen de ella los requisitos de los entrevistados y se elabora la documentación pertinente (Raghavan, Zelesnik y Ford, 1994).

La entrevista es un método para descubrir hechos y opiniones que tienen los posibles usuarios y otros participantes dentro del sistema que se está desarrollando. Posibles errores y dudas se pueden detectar desde un inicio a través de una buena entrevista. Las entrevistas pueden ser clasificadas en dos grandes grupos (Goguen y Linde, 2000):

- Las entrevistas cerradas, donde el entrevistador (ingeniero de requerimientos) prepara un conjunto de preguntas antes del encuentro con el entrevistado, y se buscan respuestas para las preguntas formuladas.
- Las entrevistas abiertas, en las cuales no se preparan preguntas concretas, y, por el contrario, se discute con el entrevistado las expectativas que este tiene del sistema.

## **Estudio de viabilidad**

El propósito del estudio de viabilidad es analizar un conjunto concreto de necesidades, con la idea de proponer una solución a corto plazo. Los resultados del estudio de viabilidad del sistema constituyen la base para tomar la decisión de seguir adelante o abandonar (Moliner, 2005).

El estudio de viabilidad se centra en los siguientes aspectos (Cortés, 1998):

- Viabilidad económica: es un análisis costo / beneficio del sistema por desarrollar.
- Viabilidad técnica: un estudio de los objetivos y el rendimiento del sistema frente a la tecnología con que se cuenta para producir los resultados que se requieren, y la posibilidad de adquirir esta tecnología en el medio.
- Recursos humanos: se determina si el personal está en la capacidad, desde el punto de vista técnico y administrativo, de llevar a cabo el proyecto.
- Viabilidad legal: es la determinación de las consideraciones y consecuencias legales del sistema.
- Viabilidad política: es la decisión de la conveniencia o no conveniencia del sistema para los grupos de poder de una organización.

Para todos los sistemas nuevos, el proceso de ingeniería de requerimientos debe empezar con un estudio de viabilidad. Los resultados del estudio de viabilidad deberían ser un informe donde se establezca si merece la pena seguir con la ingeniería de requerimientos y el proceso de desarrollo del sistema. Si el sistema no contribuye a cumplir los objetivos, entonces no tiene un valor real en el negocio (Sommerville, 2005).

### Factibilidad Económica

La factibilidad económica refiere a que la empresa debe tener la capacidad de calcular el valor de la inversión bajo evaluación antes de comprometerse a un estudio de sistemas completo. Si los costos a corto plazo no son opacados por las ganancias a largo plazo o no producen una reducción inmediata de los costos operativos, el sistema no es viable económicamente y se debe detener (Kendall y Kendall, 2005).

Se debe considerar si el proyecto es sensato o no desde el punto de vista financiero, y si los beneficios que se pronosticaron compensarán el costo y tiempo necesarios para obtenerlos. Pueden incluirse análisis de flujos de caja, como el del

valor presente que consiste en la diferencia de los ahorros que generará el proyecto menos los gastos relacionados con el proyecto mismo, después de considerar el costo financiero y el paso del tiempo (Stair y Reynolds, 2000).

#### Factibilidad Técnica

La factibilidad técnica consiste en que se puede adquirir o desarrollar el hardware, software y otros componentes de sistema para solucionar el problema (Stair y Reynolds, 2000), por lo que el analista debe averiguar si es posible actualizar o incrementar los recursos técnicos actuales de tal manera que satisfagan los requerimientos bajo consideración ya que en ocasiones los “agregados” al sistema existente son costosos y no redituables, en ese caso se debe buscar tecnología disponible que cumpla con las especificaciones. (Kendall y Kendall, 2005).

#### Factibilidad Operativa

En cuanto a la factibilidad operativa hay que evaluar si el sistema funcionará y será utilizado una vez que esté instalado. Si los usuarios están contentos con el sistema actual y no tienen problemas con su manejo habrá una fuerte resistencia a la implementación del nuevo sistema. De lo contrario si los usuarios han expresado la necesidad de que el sistema funcione mejor existe una probabilidad de que el sistema desarrollado sea utilizado (Kendall y Kendall, 2005).

### **3.2.2 Fase de Planificación**

Será necesario tener reuniones de planificación entre el equipo de trabajo, ya que según la entrevista se definirá la prioridad de los entregables, con lo que se tendrá que elaborar un cronograma de las entregas (Figura 19). Se priorizan las historias de usuario y se acuerda el alcance de los entregables. Los programadores estiman cuánto esfuerzo requiere cada historia y a partir de allí se define el cronograma. (Paulk, 2001).

## Cronograma

El cronograma consiste en un esquema que presenta las actividades a realizarse en función del tiempo estimado para culminar el trabajo. El cronograma se elabora con el diagrama de Gantt o algún Software que diseñe esta presentación. Es recomendable ilustrarlo lo más sencillo posible y señalar las actividades principales, sin las subdivisiones de cada etapa. Algunas instituciones hacen su planificación a partir de horas, otras utilizando semanas, quincenas, meses (Landeau, 2007).

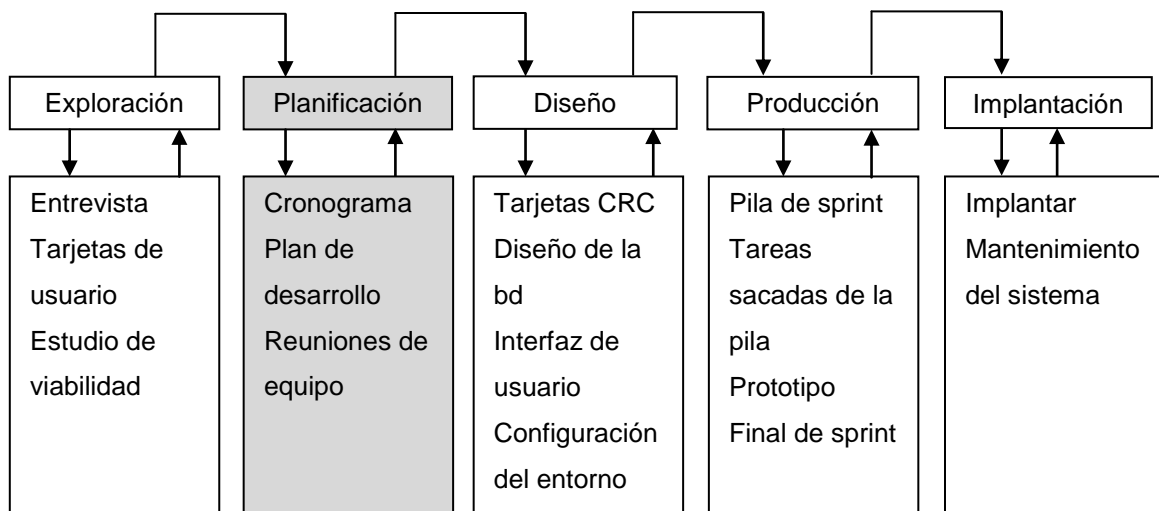


Figura 19 Fase de Planificación

## Plan de Desarrollo

La gestión efectiva de un proyecto de software depende de planificar completamente el progreso del proyecto. El gestor del proyecto debe anticiparse a los problemas que puedan surgir, así como preparar soluciones a esos problemas. Un plan, preparado al inicio de un proyecto, debe utilizarse como un conductor para el proyecto. Este plan inicial debe ser el mejor posible de acuerdo con la información disponible. Éste evolucionará conforme el proyecto progrese y la información sea mejor.

La planificación es un proceso iterativo que solamente se completa cuando el proyecto mismo se termina. Conforme la información se hace disponible, el plan debe revisarse regularmente (Sommerville, 2005).

### Reuniones de equipo de trabajo

El objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas y soluciones. En la mayoría de estas reuniones, gran parte de los participantes simplemente escuchan, sin tener mucho que aportar. Para no quitar tiempo innecesario del equipo, se sugiere realizar estas reuniones en círculo y de pie (Joskowicz, 2008).

### 3.2.3 Fase de Diseño

Se definirá la configuración del entorno de acuerdo y la interfaz de usuario además de realizar el diseño de la BD de acuerdo a las **Tarjetas CRC** (Figura 20).

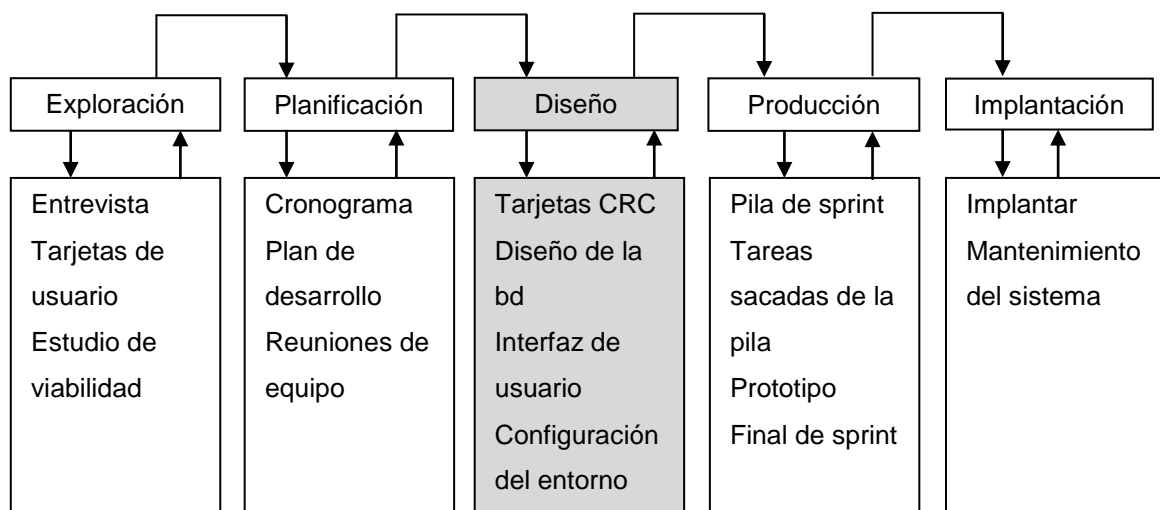


Figura 20 Fase de Diseño

Las **Tarjetas de Clase Responsabilidad-Colaboración** llamadas **CRC** representan explícitamente múltiples objetos simultáneamente. Toda la información

de un objeto se escribe en una ficha de 4x6 pulgadas. Estas tienen la ventaja de que son baratas, portátiles, de fácil acceso y familiares. En una tarjeta ideal el nombre de la clase aparece subrayado en la esquina superior izquierda, una lista de responsabilidades aparecen a la izquierda en las dos terceras partes de la tarjeta y la lista de colaboradores en el tercio derecho (Beck, 1989).

Las tarjetas están colocadas de tal manera que Vista y Controlador se superponen y se coloca por encima de modelo. Los refinamientos de una abstracción pueden ser recogidos y tratados como una sola pila de tarjetas con la tarjeta más abstracta en la parte superior donde se puede representar al resto (Beck, 1989).

El formato físico de las tarjetas CRC facilita la interacción entre los participantes del proyecto, en sesiones en las que se aplican técnicas de grupos como tormenta de ideas o juego de roles, y se ejecutan escenarios a partir de especificación de requisitos, historias de usuarios o casos de uso. Luego en un estudio de diseño avanzado o ya en la implementación del sistema, las tarjetas CRC se convierten en clases con métodos, atributos, relaciones de herencia, composición o dependencia (Casas y Reinaga, 2009).

## **Diseño de la Base de Datos**

Descompondremos el diseño de las bases de datos en tres etapas (Franch):

- 1) Etapa del diseño conceptual: en esta etapa se obtiene una estructura de la información de la futura BD independiente de la tecnología que hay que emplear. No se tiene en cuenta todavía qué tipo de base de datos se utilizará. El resultado de esta etapa se expresa mediante algún modelo de datos de alto nivel. Uno de los más empleados es el modelo entidad- interrelación.
- 2) Etapa del diseño lógico: en esta etapa se parte del resultado del diseño conceptual, que se transforma de forma que se adapte a la tecnología que se

debe emplear. Es preciso que se ajuste al sistema gestor de base de datos con el que se desea implementar la base de datos.

- 3) Etapa del diseño físico: en esta etapa se transforma la estructura obtenida en la etapa del diseño lógico, con el objetivo de conseguir una mayor eficiencia. Se deben tener en cuenta las características de los procesos que consultan y actualizan la base de datos, como por ejemplo los caminos de acceso y las frecuencias de ejecución; esto con el objetivo de conseguir un buen rendimiento de la base de datos.

### **Diseño de la interfaz de usuario**

El diseño de la interfaz de usuario (UI) es un proceso iterativo donde los usuarios interactúan con los diseñadores y prototipos de la interfaz para decidir las características, organización, apariencia y funcionamiento de la interfaz de usuario del sistema. Antes de que empiece la programación se ha de haber desarrollado, e idealmente, probado algunos diseños en papel (Sommerville, 2005).

La interfaz de usuario es en la mayoría de los casos el componente más crítico del sistema. Usuarios y operadores generalmente no entienden sobre el mundo interno de las computadoras. Conocen el sistema por medio de su interfaz, el texto, imágenes o sonidos que aparecen en los sistemas de salida del sistema. En pocas palabras, en el mundo real, para el usuario la interfaz es el “sistema” (Cañas, Granollers y Lorés, 2005).

Existen dos conceptos básicos que hay que tener en cuenta a la hora de diseñar cualquier interfaz (Cañas, Granollers y Lorés, 2005):

- 1) Visibilidad, para poder hacer una acción sobre un objeto debe ser visible.
- 2) Comprensión intuitiva, el objeto debe evidenciar en qué parte tenemos que hacer la acción y cómo la tenemos que hacer

### 3.2.4 Fase de Producción

La Fase de Producción (Figura 21) básicamente toma las fases de SCRUM donde se realiza el sprint generalmente entre 2-4 semanas y pasa a Mantenimiento, SCRUM fue desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle.

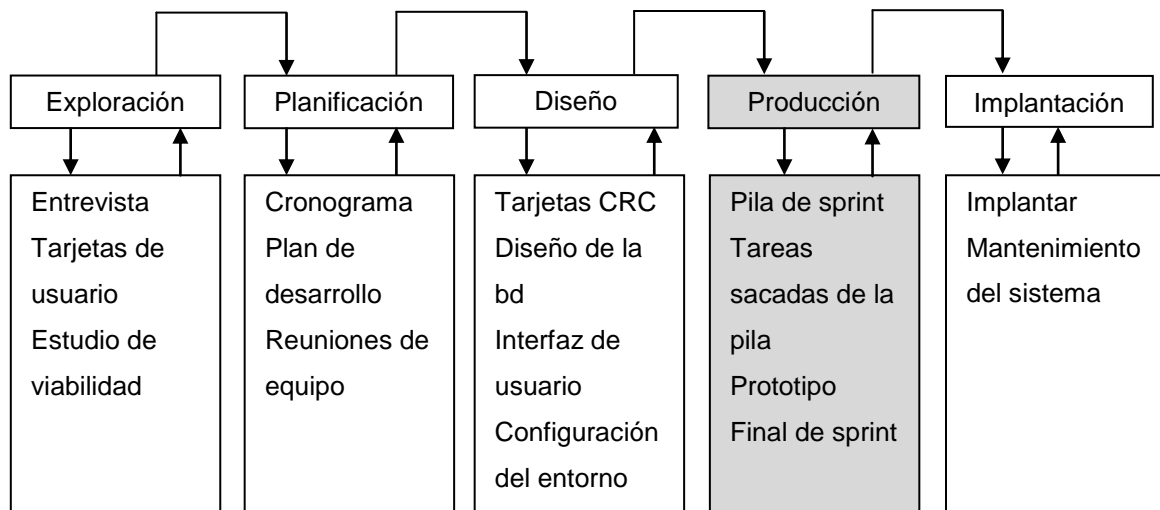


Figura 21 Fase de Producción

Scrum define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos (Schwaber, Beedle y Martin, 2001):

- El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días (Figura 22). El resultado de cada sprint es un incremento ejecutable que se muestra al cliente.
- La segunda característica importante son las reuniones a lo largo de proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración



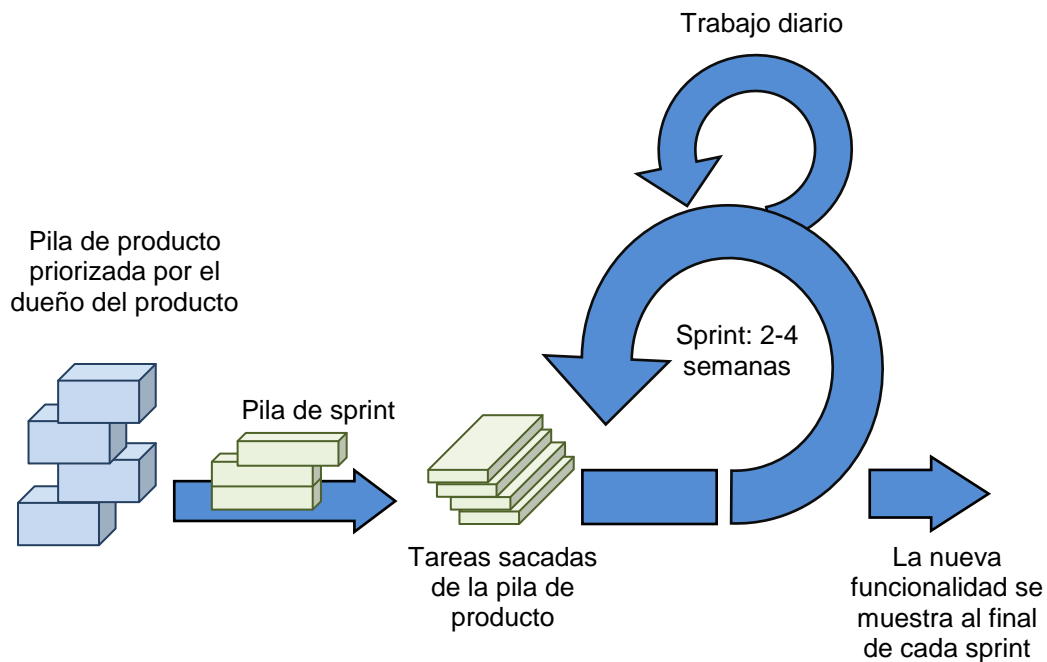


Figura 22 Desarrollo en Scrum

### Pila de Producto

El primer paso en Scrum consiste en que el dueño de producto articule la visión del producto. Al final esto evolucionará hacia una lista priorizada de funcionalidades llamada la Pila de Producto. Esta pila existe y evoluciona a lo largo de la vida del proyecto. Sólo existe una única Pila de Producto. Esto significa que el dueño de producto debe decidir sobre la priorización de todo el espectro. Al comienzo de cada Sprint se hace la Reunión de Planificación del Sprint. Se divide en dos reuniones (Deemer, ,Benefield, Larman y Vodde, 2009):

- Planificación del Sprint Parte Uno, en esta reunión el Dueño de Producto y Equipo revisan los elementos de alta prioridad de la Pila de Producto que el Dueño del producto (DP) está interesado en implementar para este Sprint. Hablan sobre los objetivos y el contexto de dichos elementos de alta prioridad

de la Pila de Producto. El DP y Equipo también revisan lo que los elementos deben cumplir.

- Planificación del Sprint Parte Dos se centra en la planificación detallada de tareas para saber cómo implementar los elementos que el equipo decide hacer. El Equipo selecciona los elementos de la Pila de Producto a las que se comprometen que estará al final del Sprint, comenzando por la parte de arriba de la Pila de Producto (en otras palabras, comenzando con los elementos que tienen más prioridad para el DP) y escribiendo la lista en orden.

### **Final de sprint**

Uno de los principios fundamentales de Scrum es que nunca se prolonga la duración del Sprint, termina en la fecha asignada aunque el equipo no haya terminado el trabajo comprometido, normalmente los equipos se comprometen en exceso en su primer Sprint y no cumplen sus objetivos. Alrededor del tercer o cuarto Sprint, los equipos suelen haber averiguado de que son capaces de entregar y cumplirán los objetivos del Sprint con más fiabilidad. A los equipos se les recomienda que escojan una duración para el Sprint y que no la cambien. Una duración consistente ayuda al equipo a saber cuánto pueden hacer, que les ayuda en la estimación y en la planificación de la entrega a más largo plazo (Deemer, Benefield, Larman y Vodde, 2009).

### **Revisión del Sprint**

Cuando termina el Sprint se hace la Revisión del Sprint, donde el equipo revisa el Sprint junto con el Dueño de Producto. Una idea clave en Scrum es inspeccionar y adaptar. Ver y aprender lo que está pasando y entonces evolucionar basándose en los comentarios y observaciones, en ciclos repetidos. La Revisión del Sprint es una actividad de inspección y adaptación del producto. Es la oportunidad de que el Dueño de Producto vea lo que está pasando con el producto y con el equipo y la oportunidad del equipo de saber cómo va el Dueño de Producto y el mercado (Deemer ,Benefield, Larman y Vodde, 2009).

### 3.2.5 Fase de Implantación

En la Fase de Implantación (Figura 23) se entrega el producto al cliente y se implanta, el mantenimiento del sistema será según lo acordado con el cliente. En esta etapa se escribe la documentación necesaria y ya no se realizan más cambios a la arquitectura, diseño o código elaborado (Paulk, 2001).

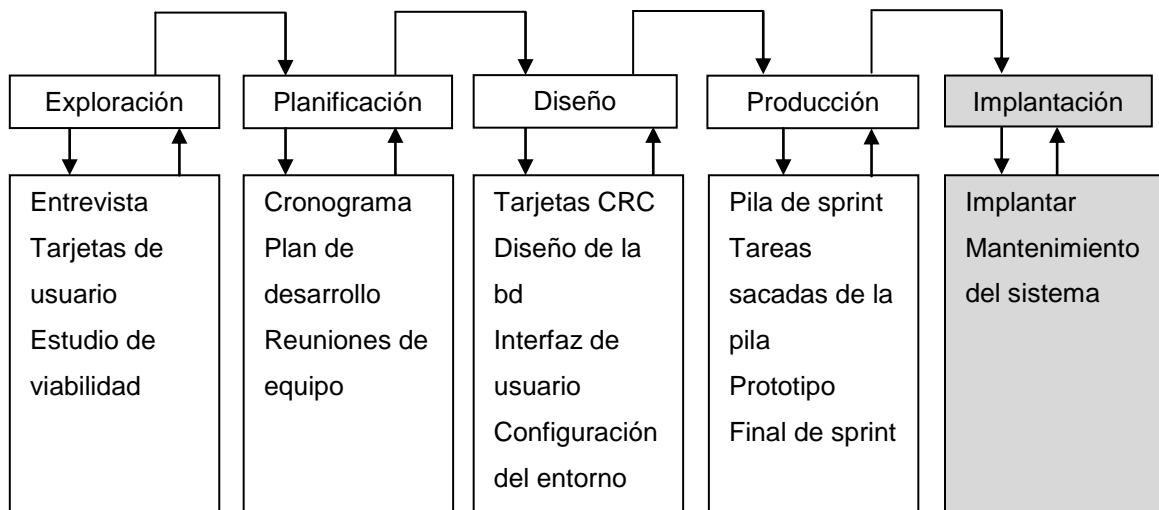


Figura 23 Fase de Implantación

### Implantación

Las pruebas de implantación se realizan con el software ya instalado en su entorno de operación real y, por tanto, se centran en dos aspectos: por una parte, los errores derivados de la integración de hardware y software del propio sistema; por otra, la interacción del sistema con el resto de sistemas de la instalación, como se muestra en la Figura 23.

En concreto, se realizan pruebas de seguridad para verificar si existen errores en los mecanismos de protección dependientes del entorno de operación. También deben realizarse pruebas de rendimiento reales, puesto que en las pruebas de sistema sólo habrán podido comprobarse rendimiento teóricos sobre el entorno de desarrollo (Tuya, Ramos y Dolado, 2007).

## **Mantenimiento del sistema**

El software debe ser mantenido, ya que sufrirá cambios después de que se entregue al cliente. Los cambios ocurrirán debido a (Alonso y Martínez, 2005):

- a) Errores encontrados (“mantenimiento correctivo”)
- b) Cambios en el entorno externo al que el software debe adaptarse, por ejemplo, la incorporación de un nuevo sistema operativo (“mantenimiento adaptativo”).
- c) Que el cliente requiere ampliaciones funcionales o desea incrementar su rendimiento (“mantenimiento perfectivo”)

Esta fase comporta diferentes actividades: Por un lado, comprobar que toda la documentación está disponible y es adecuada para las tareas de mantenimiento. Y por otro, establecer un esquema de acciones para el caso de error o modificación del software y comunicar al usuario estas acciones (Alonso y Martínez, 2005).

Los factores clave que distinguen el desarrollo y el mantenimiento, y que conducen a costes de mantenimiento, y que conducen a costes más elevados, son (Sommerville, 2005):

1. Estabilidad del equipo. Después de entregar un sistema, es normal que el equipo de desarrollo se disuelva y la gente trabaje en nuevos proyectos. El nuevo equipo o los responsables del mantenimiento del sistema no comprenden dicho sistema o las razones de fondo de las decisiones de su diseño.
2. Responsabilidad contractual. El contrato para mantener un sistema normalmente está separado del contrato para desarrollar el sistema. El contrato de mantenimiento puede darse con una compañía diferente en lugar de con el desarrollador original del sistema, este factor junto con la ausencia

de estabilidad del equipo implica que no existe incentivo para que un equipo de desarrollo escriba el software para que sea fácil de cambiar.

3. Habilidades del personal. A menudo el personal no tiene experiencia y no está familiarizado con el dominio de la aplicación. El mantenimiento tiene una pobre imagen entre los ingenieros de software.

### **Cierre**

El Documento de Aceptación se entrega al cliente y se firma por ambas partes donde de acuerdo al contrato se especifican el mantenimiento del sistema y la conformidad del cliente con el producto entregado.

## **CAPÍTULO IV. ESPECIFICACIÓN DE PROCESOS CONFORME A MOPROSOFT.**

### **4.1 Especificación de procesos para la categoría de Gerencia**

#### **4.1.1 Gestión de Proyectos**

En el **Plan Estratégico** se deben establecer los siguientes puntos (Norma NMX-I-059/02-NYCE-2005):

- Misión: El porqué de ser de la organización, la razón por la que fue creada.
- Visión: Como se visualiza la organización dentro del corto, mediano y largo plazo.
- Valores: Virtudes y cualidades deseadas y que se buscan mantener en lo trabajadores de la organización.
- Objetivos: Resultados buscados para cumplir con la Misión y Visión establecidos.
- Estrategias: Formas de lograr objetivos.
- Procesos requeridos: Identificar los procesos con su respectivo propósito, objetivos, e indicadores, estos llevarán a la organización a cumplir con las estrategias.
- Cartera de proyectos: Es el conjunto de proyectos internos o externos que representan oportunidades para la organización.
- Estructura de la organización: Consiste en definir áreas y responsabilidades al interior de la organización para cumplir con las estrategias.
- Estructura de recursos: Definir, planificar y asignar recursos para cumplir con las estrategias planteadas, considerando la base del conocimiento.
- Presupuesto: Gastos e ingresos esperados por la organización en un periodo determinado.

- Periodicidad de valoración: Definir los periodos de revisión de valoración y mejora.
- Plan de Comunicación con el cliente: Definir los mecanismos por los cuales se mantendrá comunicación con el cliente.

Así mismo al realizar el **Registro del Proyecto** se deben establecer (Norma NMX-I-059/02-NYCE-2005):

- Nombre del proyecto  
Sistema Generador y Cosechador de Metadatos Dublin Core para Documentos de Tesis
- Responsable  
Ing. Luis Enrique Rosas González
- Fechas de inicio y terminación  
Ene-2009-Jul-2009
- Cliente  
Facultad de Ingeniería Mecánica y Eléctrica de la Universidad de Colima

Además de contar con un **Plan de Mantenimiento** que especificara los responsables de dar mantenimiento al sistema, en la Tabla 2, se muestra un calendario de mantenimiento de proyecto con 3 fases (Norma NMX-I-059/02-NYCE-2005).

- Calendario

Tabla 2. Mantenimiento de Proyecto

Fase	Ago	Sep	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul
Mantenimiento 1												
Mantenimiento 2												
Mantenimiento 3												

- Responsables  
Ing. Luis Enrique Rosas González
  
- Proveedores  
Ninguno

Es necesario realizar la **Descripción del Proyecto** para informar al cliente de los alcances y entregables del producto, así como las restricciones, el plan tiene los siguientes puntos:

- Propósito  
Desarrollar un sistema generador de metadatos Dublin Core basado en un esquema de datos XML, cuya finalidad sea intercambiar información relacionada con las tesis de la comunidad universitaria con diferentes instituciones y dependencias educativas alrededor del mundo a través de diferentes bibliotecas digitales en la Web.
  
- Productos



Se obtendrá un sistema donde se podrá intercambiar y consultar información relacionada con las tesis de la universidad y diferentes instituciones educativas.

- **Objetivos**

Crear un repositorio de documentos de tesis digitales para permitir el acceso en línea a los usuarios de la comunidad universitaria, así como también intercambiar información relacionada con las tesis desarrolladas en diferentes instituciones y dependencias educativas alrededor del mundo de manera abierta y de forma libre.

- **Alcance**

El sistema podrá generar y almacenar metadatos de tesis correspondientes a la Facultad de Ingeniería Mecánica y Eléctrica, así como de instituciones educativas asociadas a la misma. A futuro se planea poder implantar el sistema en toda la Universidad de Colima.

- **Entregables**

Se realizará una única entrega al cliente de acuerdo al contrato, dicho proyecto contará con las revisiones previas sobre el desarrollo del producto, el cliente deberá dar su aprobación para la entrega del producto.

- **Necesidad de negocio**

Actualmente en la organización se realiza software a la medida de acuerdo a las necesidades del cliente, existe la necesidad de ampliar los alcances de los programadores y mediante el cumplimiento del proyecto esperan ampliar lo que saben relacionando lo desarrollado con los conocimientos de programación previos, lo que permitirá a la organización ofrecer al cliente un área de desarrollo mayor.

- Supuestos y premisas

La totalidad de los autores de tesis estarán interesados en agregar la información de sus tesis al sistema.

Los autores cumplirán a tiempo al proporcionar la información de sus tesis.

- Restricciones

Los autores de tesis deben ingresar sus datos personales y de tesis al sistema.

#### 4.1.2 Gestión de Procesos

A partir de los Procesos Requeridos, identificados en el **Plan Estratégico**, se establece o se actualiza un **Plan de Procesos**, el cual contiene los siguientes elementos (Norma NMX-I-059/02-NYCE-2005):

- **Calendario:** En la Tabla 3 se muestra el calendario para actividades en la definición, implantación y mejora de procesos incluyendo responsables.

Tabla 3 Calendario de Procesos

	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
Definición												
Implantación												
Mejora												
Responsable:	Ing. Luis Enrique Rosas González											

- **Plan de evaluación:** Las evaluaciones internas las realizarán los propios desarrolladores al finalizar cada sprint de desarrollo. Las evaluaciones externas serán realizadas por parte del cliente, es decir, la Facultad de

Ingeniería Mecánica y Eléctrica, evaluarán lo previamente acordado según el cronograma de desarrollo definido por ambas partes, se deberán presentar avances por parte de los desarrolladores sin exceder los 15 días hábiles entre cada avance y el tiempo de revisión por parte del cliente no debe exceder los 5 días hábiles.

- **Plan de manejo de riesgos:** Contiene la identificación y evaluación de riesgos, así como los planes de contención y de contingencia correspondientes. La Identificación de Riesgos determina qué riesgos pueden afectar al proyecto y documenta sus características. Entre las personas que participan en actividades de identificación de riesgos se puede incluir: el director del proyecto, miembros del equipo de proyecto, otros directores de proyectos, interesados y expertos en gestión de riesgos (PMBOK, 2005).

La Identificación de Riesgos es un proceso iterativo porque se pueden descubrir nuevos riesgos a medida que el proyecto avanza a lo largo de su ciclo de vida. Existen diversas formas de identificación de riesgos como información publicada, bases de datos comerciales, estudios académicos, estudios comparativos, o proyectos anteriores y lecciones aprendidas (PMBOK, 2005).

Se puede realizar una revisión de la documentación del proyecto, la calidad de los planes, así como la consistencia entre estos con los requisitos pueden ser indicadores de riesgos. Algunos ejemplos de técnicas utilizadas para identificar riesgos son: tormenta de ideas, entrevistas, identificación de la causa, análisis de debilidades, amenazas, fortalezas y oportunidades (DAFO) (PMBOK, 2005).

La organización puede definir escalas de impacto de ciertos objetivos del proyecto y asignarles una escala numérica de acuerdo al riesgo que representan en caso de no cumplirse, como se muestra en la Tabla 4.

Tabla 4 Definición de escalas de impacto

Objetivo del Proyecto	Escalas relativas o numéricas				
	Muy bajo/5%	Bajo/10%	Moderado/20%	Alto/40%	Muy alto/60%
Coste	Aumento del coste < 5%	Aumento del coste 5-10%	Aumento del coste 11-20%	Aumento del coste 21-33%	Aumento del coste > 34 %
Tiempo	Aumento del tiempo < 5%	Aumento del tiempo 5-14%	Aumento del tiempo 15-24%	Aumento del tiempo 25-39%	Aumento del tiempo > 40 %
Alcance	Disminución no perceptible	Áreas secundarias afectadas	Áreas principales afectadas	Reducción inaceptable para el cliente	Elemento terminado inservible
Calidad	Degradación imperceptible	Solo aplicaciones muy exigentes se ven afectadas	Reducción bajo aprobación del cliente	Reducción inaceptable por el cliente	Elemento terminado inservible

Inclusive pueden definir el impacto que tendría al proyecto otras causas relativas al personal, o ajenas a la organización, como se muestra en la Tabla 5.

Tabla 5 Otros impactos de riesgo en el proyecto

Clasificación	Escala numérica representando riesgo de incumplimiento de proyecto				
	0-10%	11-20%	21-33%	34-40%	> 41%
Inasistencia del personal	1-3 días	3-4 días	5-6 días	7-8 días	Más de 9 días
Renuncia de personal	1 elemento		2 o más elementos		
Número de visitas al cliente	> 5	3-4	2		1
Causa ajena a la organización	Corte de electricidad breve	Ataque de virus informáticos en los equipos	Aumento en costos de producción	Atraso en cumplimiento de pagos	Perdida del establecimiento por Terremoto, Inundación o Desastre natural.  Alerta epidemiológica

Dentro del desarrollo del producto se debe realizar una **Documentación de Procesos**, cada uno de ellos con los siguientes puntos (Norma NMX-I-059/02-NYCE-2005):

- Nombre del proceso  
Agregar tesis
- Descripción

El usuario introducirá al sistema los campos indicados de su tesis, como autores, tema, fecha, entre otros.

- Objetivos

El sistema generará el metadato correspondiente a la tesis para la posterior consulta de la misma.

- Indicadores

El sistema indicará los campos a ingresar

- Metas cuantitativas

Total de campos solicitados

- Responsabilidad y autoridad

Ing. Luis Enrique Rosas González

- Procesos relacionados

Consultar tesis, Eliminar tesis, Modificar tesis

- Entradas

Autor, Tema, Área, Fecha, Lugar, Asesor.

- Salidas

No

- Productos internos

Metadato de tesis

- Roles involucrados

Desarrollador

- Actividades  
Ingresar datos al sistema
- Verificaciones y validaciones  
El sistema invalidará la alta de la tesis cuyos datos estén incompletos
- Incorporación a la base de conocimiento  
Dar de alta información
- Recursos de infraestructura  
Equipo de cómputo con características iguales o superiores al equipo donde fue implementado el sistema por primera vez.
- Mediciones  
Totalidad de datos completados
- Situaciones excepcionales  
Autor no disponible para ingresar datos al sistema
- Lecciones aprendidas  
La información agregada al sistema deberá estar completa.
- Guías de ajuste  
No

### 4.1.3 Gestión de Recursos

Dentro de los recursos de la organización también se tiene un Registro **de Recursos Humanos**, para el ejemplo se muestran los datos del encargado del proyecto (Norma NMX-I-059/02-NYCE-2005):

- Datos personales  
Nombre: Luis Enrique Rosas González  
Edad: 24 años  
Ciudad de Origen: Armería, Colima
  
- Formación  
Ing. Sistemas Computacionales, Universidad de Colima
  
- Experiencia  
2 años desarrollando sistemas para microempresas. Dominio de lenguaje PHP, .NET, C##, CSS, JavaScript, Ajax.
  
- Roles asignados  
Gerente
  
- Evaluaciones de desempeño.  
Evaluación satisfactoria (May-2009)

## 4.2 Especificación de procesos para la categoría de Operación

### 4.2.1 Fase de Exploración

- Entrevista



En la entrevista es clave el trato con la persona, se sugiere utilizar un formato mixto de preguntas abiertas y cerradas, después de una breve presentación con el cliente por parte del entrevistador, algunas preguntas pueden ser:

- 1.- ¿Cuáles son las causas que le motivaron a buscar alguien que desarrolle Software?
- 2.- ¿Es usted la persona con la que se mantendrá contacto durante el desarrollo del proyecto?
- 3.- En cuanto a la disponibilidad de horario, ¿cuándo se podrían realizar las reuniones y durante cuánto tiempo?
- 4.- ¿Quiénes serían los usuarios?
- 5.- ¿En qué equipos utilizaría el Software desarrollado?
- 6.- ¿Cuál es la actividad en que sería utilizado el Software?
- 7.- ¿Existe algún Software similar?
- 8.- ¿Es necesario algún lenguaje de programación en específico en el cual deba ser desarrollado el proyecto?
- 9.- ¿Cómo serían dados de alta los usuarios?
- 10.- ¿Existe algún logotipo o colores que deban ser utilizados en el diseño?
- 11.- ¿Cómo sería presentada la información al usuario?
- 12.- Mencione los campos de la base de datos utilizados en el Software
  
- **Tarjetas de usuario**

En la Figura 24 se muestra el anverso de una tarjeta de usuario y en la Figura 25 el reverso de la misma.

### Anverso

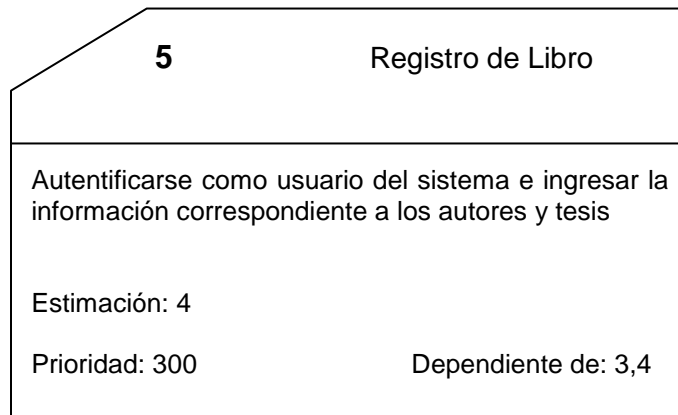


Figura 24 Anverso de tarjeta de usuario

### Reverso

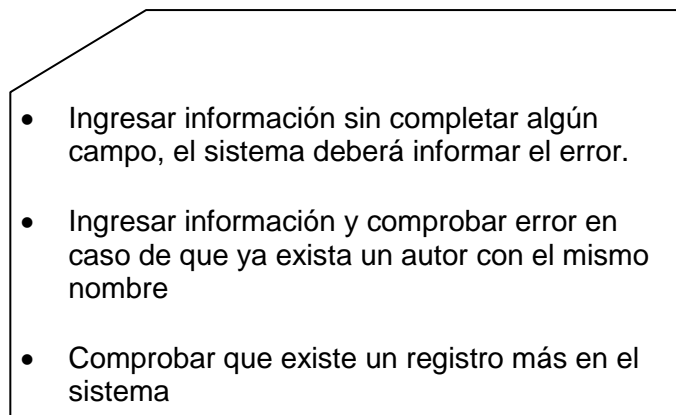


Figura 25 Reverso de tarjeta de usuario

- **Estudio de viabilidad**

Para realizar el estudio de viabilidad técnica se realizó una recolección de información sobre la tecnología disponible al interior de la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad de Colima y la posibilidad de hacer uso de dicha tecnología para el desarrollo e implementación del sistema a desarrollar y en

caso de ser necesario plantear los requerimientos tecnológicos que se debieran adquirir para el desarrollo y puesta en marcha. De acuerdo a la tecnología necesaria para la implantación la evaluación se realizó sobre dos enfoques: **Hardware y Software**.

Respecto al **Hardware**, específicamente el servidor donde quedará instalada la plataforma debe contar los siguientes requerimientos mínimos:

- Procesador Intel Pentium 4 a 3.2 Ghz
- Tarjeta Madre
- 4 GB de Memoria RAM
- Disco Duro de 500 GB
- Unidad de CD-ROM
- Tarjeta de Red
- Tarjeta de Video
- Monitor SVGA
- Teclado
- Mouse

Evaluando el hardware existente y tomando en cuenta la configuración mínima para la operación de la plataforma la institución no requirió realizar una inversión inicial para la adquisición de equipos de cómputo, tampoco actualizar los equipos existentes, dado que los mismos satisfacen los requerimientos mínimos para el desarrollo y puesta en marcha de la plataforma.

Debido a las características del sistema es necesario contar en cada área con una red interna, misma que permita la interconexión de todos los equipos de la institución. Para aquellos equipos que por su ubicación geográfica no puedan hacer uso de la Red cableada se prevé realizar la conexión vía inalámbrica.

Las características de la red interna con la que se cuenta actualmente se detallan a continuación:

- Servidor: Equipo con procesador Intel Xeon de cuatro núcleos, 16 GB de Memoria Ram DDR3, Disco Duro de 4 Tb, Tarjeta de Red 100/1000.
- Estaciones de Trabajo: Equipo con procesador Intel Dual Core i3, 2GB de Memoria Ram DDR3, Disco Duro de 250 Gb, Tarjeta de Red 100/1000.
- Concentradores de Puertos RJ-45

Todas las estaciones de trabajo están conectadas al servidor a través de una red de topología estrella, utilizando cable par trenzado de la categoría número cinco.

En cuanto al **Software**, la Institución ya cuenta con las aplicaciones empleadas en el desarrollo del proyecto y en el funcionamiento del sistema, por lo cual no fue necesaria inversión alguna. Las estaciones de trabajo y el servidor operan bajo sistema Operativo Windows 7.

#### Viabilidad Operativa

Existe la necesidad de implementar una plataforma. Con la finalidad de garantizar un buen funcionamiento de la plataforma y un impacto positivo en los usuarios la plataforma fue desarrollada siguiendo los lineamientos de la institución, presentando al usuario una interfaz amigable lo que hace de ella una plataforma de manejo fácil y rápida comprensión, además de cubrir todas las necesidades, requerimientos y objetivos plantados. Esto demuestra que no se presenta ninguna oposición para llevarse a cabo, por lo que el sistema es factible operacionalmente. En el proceso de adiestramiento se detallarán los aspectos de actualización de conocimientos y

nuevas formas en el procesamiento de la información que representa el manejo de una plataforma.

#### Viabilidad Legal

El desarrollo del sistema se realizará en lenguaje PHP, es un software de tipo libre en el que no hay que pagar licencia, por lo que no se incurre en ninguna falta. Para la creación de la base de datos, el sistema recurre a la utilización de MySQL que es un sistema de gestión de base de datos relacional multiusuario, perteneciente a SUN Microsystems, con un esquema de licenciamiento de software libre, por lo tanto no se requiere de la adquisición alguna licencia. Se dedujo utilizar software libre para evitar problemas con las licencias, por lo cual dadas las se puede decir que el desarrollo del sistema es legalmente factible.

#### 4.2.2 Fase de Planificación

- **Cronograma**

En la Tabla 6 se muestra un cronograma del desarrollo del proyecto bajo el modelo propuesto.

Tabla 6 Desarrollo del Proyecto

Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Exploración																	
Entrevista																	
Tarjetas de usuario																	
Estudio de																	

viabilidad																			
Planificación																			
Cronograma																			
Plan de desarrollo																			
Reuniones de equipo																			
Diseño																			
Tarjetas CRC																			
Diseño de la BD																			
Interfaz de usuario																			
Configuración del entorno																			
Producción																			
Pila de Sprint																			
Tareas sacadas de la pila																			
Prototipo																			
Final de sprint																			
Implantación																			



Consulta	
<ul style="list-style-type: none"> <li>- Consultar la tesis</li> <li>- Consultar información de autores</li> <li>- Consultar información de la tesis</li> </ul> <p>Desplegar información de autores</p> <p>Desplegar información de tesis</p>	<p>Registro</p> <p>Modificar</p>

Figura 27 Tarjeta CRC 2

Búsqueda	
<ul style="list-style-type: none"> <li>- Buscar información registrada</li> <li>- Ingresar una cadena de texto</li> </ul> <p>Desplegar los resultados de la búsqueda</p>	<p>Consulta</p>

Figura 28 Tarjeta CRC 3

Modificar	
<ul style="list-style-type: none"> <li>- Modificar información de los autores</li> <li>- Modificar información del documento de tesis</li> </ul> <p>Verificar habilitación de usuario</p> <p>Actualizar información</p>	<p>Registro</p> <p>Consulta</p>

Figura 29 Tarjeta CRC 4



#### **4.2.4 Fase de Implantación**

- **Documento de aceptación**

El documento de aceptación se pone a consideración del cliente, deberá ser un documento donde el cliente especifique que queda conforme con el producto entregado, en cuanto a funcionamiento, diseño y diversos aspectos previamente negociados con la microempresa, también debe mencionarse el mantenimiento y las condiciones en las que se realizarán si así procede, así como la capacitación para los usuarios finales.

## CAPÍTULO V PRUEBAS Y RESULTADOS

En este capítulo se presentan comparativas de las fases entre diferentes metodologías ágiles tomando en cuenta los tiempos recomendados por los autores para cada etapa.

**Extremme Programming (XP)** (Wells, 1999) se muestra en la Tabla 7.

Tabla 7. Fases de XP

Fase	Tiempo de la subfase	Tiempo estimado de la fase	Tiempo acumulado
Exploración		14-23 días	
<i>Historias de usuario</i>	7- 16 días		
<i>Plan de publicaciones</i>	4 días		
<i>Plan de iteraciones</i>	3 días		14-23 días
Planeamiento		4 días	
<i>Cronograma</i>	2 días		18-27 días
Diseño		4 días	
<i>Tarjetas crc</i>	2 días		22-31 días

El tiempo estimado para las primeras tres fases antes del desarrollo es de entre 22-31 días debido a las Tarjetas de historias de usuario.

**DSDM** (Stapleton, 1997), se muestra en la Tabla 8.

Tabla 8. Fases DSDM

Fase	Tiempo de la subfase	Tiempo estimado de la fase	Tiempo acumulado
Estudio de viabilidad	7- 12 días	7- 12 días	7- 12 días
<i>Estudio de negocio</i>	7- 12 días	7- 12 días	14- 24 días

Iteración de modelado funcional		6- 8 días	
<i>Acordar calendario</i>	2 días		
<i>Crear prototipo de funcionalidad</i>	2-4 días		
<i>Revisar prototipo</i>	2 días		20-22 días
Iteración de diseño y desarrollo		6- 8 días	
<i>Acordar calendario</i>	2 días		
<i>Crear prototipo de diseño</i>	2-4 días		
<i>Revisar prototipo</i>	2 días		26-28 días

El tiempo estimado para las primeras tres fases, antes de iniciar el desarrollo es de entre 26 y 28 días estimados, tomando en cuenta el mismo proyecto bajo la metodología XP.

**Metodología propuesta**, se muestra en la Tabla 9.

Tabla 9. Metodología propuesta.

Fase	Tiempo de la subfase	Tiempo estimado de la fase	Tiempo acumulado
Exploración		15-28 días	
<i>Entrevista</i>	1 día		
<i>Historias de usuario</i>	7-16 días		
<i>Estudio de viabilidad</i>	7-12 días		15- 28 días
Planificación			
<i>Cronograma</i>	2 días	4 días	
<i>Plan de desarrollo</i>	2 días		19-32 días
Diseño			
<i>Tarjetas crc</i>	2 días	6 días	

<i>Diseño de la bd</i>	2 días		
<i>Interfaz de usuario</i>	2 días		25-38 días

De acuerdo a los tiempos estimados para las metodologías se obtuvo que la metodología XP, obtuvo un menor tiempo en sus primeras tres fases, sin embargo no cuenta con un estudio de viabilidad que permita saber si el proyecto es viable, la metodología DSDM obtuvo un tiempo máximo de 28 días aproximadamente, que es inferior al tiempo máximo aproximado de XP, pero no cuenta con un plan de desarrollo.

Aunque en la propuesta se obtuvo un tiempo mínimo similar a las dos metodologías previas el tiempo máximo fue mayor debido a un mayor número de subfases, pero se consideran necesarias debido a que se especifica el plan de desarrollo, el diseño de la bd, y la interfaz de usuario.

Tabla 10 Proyecto MoproSoft

MoproSoft			No. De Proceso
Alta gerencia		Reportes financieros	1
		Factores externos	2
Gerencia	Gestión de proyectos	Plan estratégico	3
		Plan de gestión de proyectos	4
		Reporte de validación	5
		Alternativas de realización de proyectos específicos	6
		Registro del proyecto	7
		Contrato	8
		Descripción del proyecto	9
	Gestión de procesos	Plan de procesos	10

		Documentación de procesos	11
	Gestión de recursos	Plan de adquisiciones y capacitación	12
		Plan operativo de conocimiento de la organización	13
		Plan operativo de bienes, servicios e infraestructura	14
		Solicitudes de bienes o servicios	15
		Plan operativo de recursos humanos y ambiente de trabajo	16
		Solicitudes de capacitación y de personal	17
		Plan de administración de la base de conocimiento	18
		Diseño de la base de conocimiento	19
		Plan de mantenimiento	20
		Catalogo de proveedores	21
		Registro de bienes o servicios	22
		Plan de capacitación	23
		Registro de recursos humanos	24

Operación	Admón. de proyectos específicos	Plan de desarrollo	25
		Plan del proyecto	26
		Documento de aceptación	27
	Desarrollo y mantenimiento de software	Especificación de requerimientos	28
		Análisis y diseño	29
		Componente	30
		Manual de operación	31
		Manual de usuario	32
		Software	33
		Configuración de software	34

Los procesos sombreados son los que se adaptaron al nuevo modelo, el tiempo considerado de 2 días en algunos documentos es sólo en la elaboración, considerando un proyecto bajo MoproSoft, Tabla 10, el tiempo empleado en él es de 179 días aproximadamente, en las mismas condiciones un proyecto con el Modelo propuesto es de 134 días como se muestra en la Tabla 11, con lo que se reduce más de un mes y la microempresa puede enfocarse en desarrollar nuevos proyectos, debido a la necesidad de estar en continuo crecimiento y tener siempre proyectos en puerta.

Tabla 11 Proyecto Modelo propuesto

Modelo propuesto			No. De Proceso
Gerencia	Gestión de proyectos	Plan estratégico	1
		Plan de gestión de proyectos	2
		Reporte de	3

		validación		
		Plan del proyecto	4	
		Descripción del proyecto	5	
		Registro del proyecto	6	
		Contrato	7	
	Gestión de procesos	Plan de procesos	8	
		Reporte de verificación y validación	9	
		Documentación de procesos	10	
	Gestión de recursos	Registro de recursos humanos	11	
		Diseño de la base de conocimiento	12	
		Plan de mantenimiento	13	
	Operación	Administración de proyectos específicos	Plan del proyecto de registro de actividades	14
			Minuta de reuniones	15
Desarrollo y mantenimiento de software		Exploración	Entrevista	16
			Tarjetas de usuario	17
			Estudio de viabilidad	18
		Planificación	Cronograma	19
			Plan de desarrollo	20
		Diseño	Tarjetas CRC	21

			Diseño de la bd	22	
			Interfaz de usuario	23	
			Configuración del entorno	24	
		Producción	Pila de sprint	25	
			Tareas sacadas de la pila	26	
			Prototipo	27	
			Final de sprint	28	
		Implantación	Implantar	29	
			Mantenimiento del sistema	30	
				Manual de operación	31
				Manual de usuario	32
				Documento de aceptación	33

Comparado a MoproSoft el Modelo propuesto adopta nuevas fases en el nivel de Operación en el Desarrollo y Mantenimiento de Software: Exploración, Planificación, Diseño, Producción e Implantación. Se decidió optar por ellas basándose en metodologías ágiles como XP, DSDM y Scrum por ser las más utilizadas en las microempresas y con el fin de permitirles una adaptación más rápida al no tener que cambiar sustancialmente la forma en que desarrollan.

En cuanto al número de procesos resultó similar debido a los cambios en el nivel de Operación donde se agregaron subfases en el Desarrollo y Mantenimiento de Software, el nuevo modelo fue probado en un sistema desarrollado para la Delegación 3 de la Universidad de Colima, el cual consiste en un Registro de



Actividades, en el que podrá agendar tareas, consultar las tareas, actualizarlas y generar los reportes correspondientes.

A continuación se muestra el documento de descripción del proyecto perteneciente al sistema Registro de Actividades, el cual fue desarrollado bajo el modelo propuesto.

## **DESCRIPCIÓN DEL PROYECTO**

### **Datos generales**

#### **Asunto o proyecto**

Registro de actividades

#### **Fecha**

12 de febrero de 2011

### **Productos**

El sistema integrará información de los eventos, las actividades, programas, proyectos, cursos, talleres de su delegación, contando con 4 módulos para su funcionamiento: Agendar tareas, Consultar por día, Actualizar tareas y Generar reportes.

### **Propósito**

El propósito del producto Registro de actividades es apoyar a la delegación 3 en el registro de sus actividades, de manera automatizada.

### **Objetivos**

Desarrollar un sistema que permita que se puedan registrar y manipular las actividades y recursos de la Delegación 3.

Cumplir con la fecha de terminación establecida por el cliente.

**Alcance**

El objetivo principal del Sistema de Registro de Actividades, es el de brindar una herramienta para el control de las tareas que la Delegación No. 3 desempeña a lo largo del año y que de una manera sencilla y rápida se generen reportes acerca de las mismas

**Entregables**

Diagrama del sistema, primer avance del sistema, diseño de la interfaz, segundo avance del sistema, entrega del sistema con diseño integrado, integración de correcciones, entrega final.

**Contacto o personas de enlace**

Eric Morfín, Delegación No. 3 Universidad de Colima

**Atendió**

Griselda Michel Ugarte

**Medio**

Personalmente

**Necesidad del Negocio**

El delegado tiene una serie de actividades que debe cumplir y después reportar continuamente, con este sistema esto será más fácil y rápido de detectar y de llevar el control.

**Observaciones, Supuestos, Restricciones y Premisas**

Servidor web con conexión a Internet, espacio de disco suficiente, PHP versión 4.3 o superior, MySQL versión 4.1 o superior, cuenta de acceso a MySQL con un usuario y contraseña sobre una base de datos (nombre “registro de actividades”).

**Fecha de entrega preliminar**

Finales de Mayo de 2011

**Estimación de costo**

\$60,000.00

**Responsable del proyecto**

Gilberto Flores Assam

**Datos de Facturación**

Dirección Av. Universidad No. 333

Col. Las Víboras

C.P. 28040

Ciudad Colima, Col.

País México

Tels. /Fax 31 61000

R.F.C UCO6208089R6

A continuación se muestra parte del documento plan del proyecto perteneciente al sistema Registro de Actividades

**Organización del Proyecto**

En la organización se definió el equipo de trabajo, Tabla 12 y las interfaces externas, Tabla 13.

Tabla 12 Definición de la estructura de equipo de trabajo

Rol	Recurso	Responsabilidad
GPY	Griselda Michel	Conocer y aprobar el Plan de proyectos
APE	José Gilberto Flores Assam	Crear y ejecutar el Plan de Proyectos
Programador	José Manuel Olivares Corona	Desarrollar el Proyecto conforme al Plan de Proyectos
Diseñador	Miguel Ángel Ávila	Desarrollar la interfaz gráfica

Tabla 13 Interfaces externas

Grupo / Empresa	Contacto	Cargo / Rol	Modo de Comunicación
Delegación No. 3	Eric Morfín	Responsable del proyecto	Telefónica / Correo Electrónico /Presencial

## Ciclos y actividades

### Ciclo de vida

En la Figura 30 se presenta el ciclo de vida del sistema. Este proyecto no contará con iteraciones, ya que el cliente ha definido exactamente los módulos que requiere, el modelo propuesto permite las iteraciones en caso de ser necesarias.

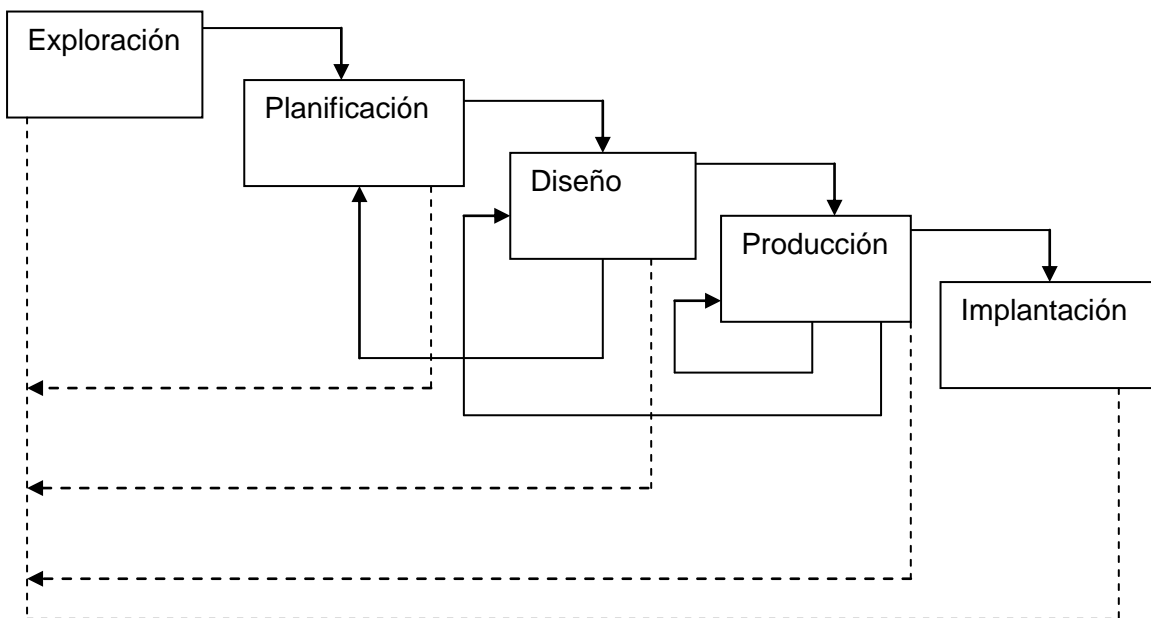


Figura 30. Ciclo de vida

## **Etapas del ciclo de vida**

**Exploración:** Se hace el primer acercamiento con el cliente para definir los requerimientos

**Planificación:** Realizar planes para la administración del proyecto.

**Diseño:** Analizar los requerimientos y realizar el diseño de la solución.

**Producción:** Realizar el producto según las especificaciones del diseño.

**Implantación:** El producto se entrega al cliente y se implanta.

## **Planificación del Proyecto**

### **Estimaciones**

Dentro del desarrollo del proyecto se estima el esfuerzo del proyecto, Tabla 14, por lo que se contempla el trabajo de las siguientes personas:

Tabla 14 Estimación de esfuerzo del proyecto

Rol	No, de personas	Días	Horas x día	Totales
Analista	3	4	4	48
Diseñador	2	10	4	80
Programador	1	36	8	288
Control de calidad	1	3	8	24
Total	7	53	24	440

El total estimado del proyecto fue de 53 días, cada día con 8 horas laborales, lo que resultó en 440 horas.

### **Estimación de costo:**

Proyecto universitario N/A.

Se definió con el cliente los protocolos de entrega, Tabla 15.

Tabla 15 Protocolo de entrega

Entregables	Protocolo
Diagrama del sistema	Presencial
Primer avance del sistema	Presencial
Diseño de la interfaz	Presencial / Envío de Información por correo electrónico
Segundo avance del sistema	Presencial
Entrega del sistema con el diseño integrado	Presencial / Disponibilidad en el servidor
Integración de correcciones (en caso de existir)	Presencial / Disponibilidad en el servidor

La entrega de documentos del sistema Registro de Actividades se realizó en las fechas especificadas en la Tabla 16:

Tabla 16 Entrega de documentos

<b>Nombre del Elemento</b>	<b>Fecha de entrega</b>
Plan Estratégico	7 de Febrero de 2011
Glosario Organizacional	Febrero de 2008 (sin cambio)
Plan de Procesos	6 de Febrero de 2011
Estándar de Documentación	Enero de 2008 (sin cambio)
Estándar de Documentación de Procesos	Enero de 2008
Documentación de Procesos (Estándares, Procesos y Plantillas)	Enero y Febrero de 2010 (sin cambio)
Plan de Gestión de Proyectos	Marzo de 2008
Registro de Proyectos	En sistema
Minuta de reuniones	Variable
Descripción del Proyecto	Variable (en plan de proyecto)

<b>Nombre del Elemento</b>	<b>Fecha de entrega</b>
Plan Operativo de Recursos Humanos y Ambiente de Trabajo	10 de Marzo de 2011
Plan Operativo de Conocimiento de la Organización	10 de Marzo de 2011
Plan de Mantenimiento	13 de Marzo 2011
Conocimiento de la Organización	
Plan de Administración de la Base de Conocimiento	13 de Marzo de 2011
Diseño de la Base de Conocimiento	13 de Marzo de 2011
Recursos Humanos y Ambiente de Trabajo	
Registro de Recursos Humanos	Mayo de 2010 (sin cambios)
Administración de Proyectos Específicos	
Plan del Proyecto de Registro de Actividades	14 de Febrero de 2011
Minuta de Reuniones	Variable
Proyecto piloto Desarrollo y Mantenimiento de Software	
Especificación de Requerimientos de Registro de Actividades	14 de Febrero de 2011 (en plan de proyecto)
Manual de Usuario	Mayo de 2011
Diseño	16 de Febrero de 2011
Manual de Operación	Mayo de 2011
Configuración de Software	Abril de 2011

Dentro de las suposiciones, dependencias y restricciones se tuvo que en CENEDIC se depende principalmente de las disposiciones de la Coordinación General de Servicios y Tecnologías de Información, plasmadas en su Programa

Operativo Anual, así como de la Rectoría quien crea el Programa Institucional de Desarrollo 2009-2013(PIDE) y otras dependencias como la Dirección General de Planeación, Dirección General de Propiedad Intelectual, etc. El plan de procesos se baso en la Planeación Estratégica establecida al inicio del 2011 en CENEDIC.

En el nivel de operación se definieron las tareas a partir del plan de trabajo, el desarrollo del sistema se dividió en módulos para la entrega con el cliente, especificando por anticipado los módulos a entregar, las tareas se realizaron en días laborales de 8 horas, los tiempos empleados fueron los que se muestran en la Tabla 17:

Tabla 17 Tiempos del nivel Operación

<b>id</b>	<b>Nombre de tarea</b>	<b>Duración</b>
1	Creación del plan de trabajo	4 días
2	Conceptualizar el sistema	1 día
3	Analizar los requerimientos	1 día
4	Definir el diseño global	1 día
5	Definir periodos	1 día
6	Realizar el diseño global de las interfaces del sistema	1 día
7	Realizar el diseño	5 días
8	Desarrollo del sistema	26 días
9	Introducción al sistema( análisis con el programador)	1 día
10	Bases de datos	1 día
11	Registro de usuario	2 días
12	Módulo del usuario	6 días
13	Acceso al sistema	1 día
14	Asignación de formas	1 día
15	Recepción de formas	2 días
16	Revisión de formas	2 días
17	Módulo del administrador	16 días
18	Acceso al sistema	1 día



19	Creación de usuarios	2 días
20	Catálogo de formas	1 día
21	Altas	1 día
22	Bajas	1 día
23	Modificación	1 día
24	Consultas	8 días
25	Tipo de documento	2 días
26	Fecha	2 días
27	No de folio	1 día
28	Asignados	1 día
29	Comprobadas	1 día
30	Sin asignar	1 día
31	Reportes	5 días
32	Pruebas del sistema	2 días
33	Control de calidad	3 días

El control de calidad se realizó con el propósito de detectar las posibles fallas o correcciones a realizar antes de entregar al cliente el producto final.

El cliente firmó de conformidad que durante el desarrollo del sistema Registro de Actividades obtuvo los entregables mostrados en la Tabla 18:

Tabla 18 Entregables del sistema

<b>Entregable</b>	<b>Fecha de compromiso</b>	<b>Observaciones</b>
Diagrama del sistema	14 de Febrero de 2011	Se le entregó al cliente la estructura en la fecha indicada para su verificación, se le mandó impreso para que se aprobara
Diseño de la interfaz	16 de Febrero de 2011	Se mandó al cliente el diseño de las pantallas en

		forma impresa para su verificación
Primer avance del sistema	17 de Marzo de 2011	Se le presentó al cliente el módulo de catálogos para verificar avances
Segundo avance del sistema	8 de Abril de 2011	Se le presentó al cliente el sistema ya casi concluido en todos sus módulos
Entrega del sistema con el diseño integrado	16 de Mayo de 2011	Se le entregó al cliente el proyecto terminado con todo y la interfaz gráfica para realizar las pruebas con la información real en la delegación
Elaboración de correcciones	23 de Mayo de 2011	Se probó el sistema y terminaron las correcciones indicadas por el cliente
Entrega final	25 de Mayo de 2011	Entrega final con todas las indicaciones que el cliente comentó

El sistema desarrollado bajo el modelo propuesto fue entregado en tiempo y forma al cliente el 25 de Mayo, 100 días después de que el cliente aprobó la estructura del sistema, el número de procesos realizados en el nuevo modelo fueron:

Nivel Gerencia: 13 procesos

Nivel Operación: 20 procesos

El sistema desarrollado bajo MoproSoft hubiera requerido del siguiente número de procesos por nivel:

Nivel Alta dirección: 2

Nivel Gerencia: 22

Nivel Operación: 10

El nivel de Alta dirección se unió al nivel Gerencia dentro del nuevo modelo, asumiendo que en la microempresa el encargado de la Alta dirección realiza también labores de Gerencia, en el nuevo modelo se redujo en 9 el número de procesos a nivel Gerencial respecto a MoproSoft y aunque en el Nivel de Operación aumentó al doble el número de procesos el total de procesos del nuevo modelo es menor respecto a MoproSoft.

Dentro del nivel de Operación se encontraron problemas en la parte del diseño, ya que los programadores tenían como práctica común realizar diagramas de casos de uso y no Tarjetas CRC como lo propone el nuevo modelo.

Se encontró también que el modelo propuesto no tenía contemplado llevar el registro de los entregables con el cliente, ni de especificar las condiciones de los mismos.

El nuevo modelo resultó fácil de entender por parte de los analistas y programadores debido a la similitud con MoproSoft y sugirieron incluir los diagramas de casos de uso en la fase de Diseño.

El nuevo modelo deja a consideración de la microempresa el plan de capacitación, asumiendo que el programador conoce el lenguaje de programación que el cliente solicita. En caso contrario se puede considerar incluirlo.

## **CAPÍTULO VI. CONCLUSIONES**

### **6.1 Discusión**

El desarrollo de este modelo de calidad para la microempresa basado en MoproSoft tiene importancia en cuanto a que toma las partes más significativas de algunas de las metodologías más utilizadas por parte de las microempresas en el desarrollo de software y sirve de base para certificarse en MoproSoft o ISO, lo que les haría más competitivas en caso de que exista alguna licitación cuyos requisitos sean la certificación en alguna de las normas anteriores. Algunos de los problemas enfrentados en el desarrollo fue la poca disponibilidad de material respecto a los requisitos necesarios para certificarse en MoproSoft, la falta de encuestas sobre el número de microempresas certificadas y la norma que adoptaron. El modelo desarrollado sólo aplica a microempresas y se sugiere que esté compuesta por al menos 5 empleados asignando roles específicos en funciones para el cumplimiento del proyecto.

### **6.2 Cumplimiento de las hipótesis de trabajo**

Considerando los resultados obtenidos en las pruebas, se logró reducir MoproSoft y adecuar a las microempresas a través de un nuevo modelo que agiliza los procesos y reduce el tiempo de trabajo comparado con una forma de trabajo sin organización, el modelo baso en las metodologías más utilizadas como Xp y Scrum. Se logro identificar las características de MoproSoft que son más viables en el ambiente de desarrollo de una microempresa debido al número de personal con el que cuentan y las dificultades que presentan, se pudo desarrollar con éxito un Sistema de Registro de Actividades bajo el nuevo modelo, por lo que se cumplió con la hipótesis de trabajo planteada en un inicio.

### 6.3 Recomendaciones

Con el propósito de continuar con la investigación en el futuro se hacen las siguientes recomendaciones:

Realizar una unión entre diferentes metodologías a las mostradas con el fin de abarcar un mayor número de microempresas y que el cambio que se realice afecte el mínimo posible a su forma de trabajo.

Realizar una implementación del modelo en microempresas con un número de empleados superior a siete y analizar el funcionamiento del modelo y las mejoras posibles.

Probar el sistema propuesto al desarrollar un sistema de mayor complejidad y número de módulos que el sistema con el que fue probado.

## REFERENCIAS

Alonso F., Loïc Martínez. Introducción a la ingeniería del software.2005. España. Pp 89

Alquicira C. y Su A. Pruebas Controladas de MoProSoft., Revista Software Guru Conocimiento en Práctica, Año 01 No.01, 2005, pp.28-30.

Ambler, S. Agile Adoption Survey 2008. Disponible en:  
<http://www.ambyssoft.com/surveys/agileFebruary2008.html> Visitado Diciembre 2010.

Astorga María, Olguín Martín, Flores Brenda. Caracterización de los factores de cambio para la implantación de MoProSoft en las PyMES. 1er. Congreso Internacional de Ciencias Computacionales (CICOMP). 2006. México. Pp. 13-17.

Beck Kent. A Laboratory For Teaching Object-Oriented Thinking. Apple Computer. OOPSLA'89 USA. Disponible en : <http://c2.com/doc/oopsla89/paper.html>

Canós José, Letelier Patricio, Penadés Ma. Carmen, Universidad Politécnica de Valencia. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Disponible en: <http://www.willydev.net/descargas/prev/TodoAgil.pdf>

Cañas José, Granollers Toni, Lorés Jesús. Diseño de sistemas interactivos centrados en el usuario. 2005. España. Pp 24-25

Casas Sandra, Reinaga Héctor. Aspectos Tempranos: un enfoque basado en Tarjetas CRC. Cuarto Congreso Colombiano de Computación 4CCC. 2009. Colombia. Disponible en: <http://espanol.oocities.com/profeprog2/INVPAPER31.pdf>

Calderón Amaro, Dámaris Sarah, Rebaza Valverde, Carlos Jorge. Universidad Nacional de Trujillo. Metodologías Ágiles.2007. Perú. Disponible en: [www.seccperu.org/files/Metodologias%20Agiles.pdf](http://www.seccperu.org/files/Metodologias%20Agiles.pdf)

Cortés Roberto. Introducción al análisis de sistemas y la ingeniería del software. 1998. Costa Rica. Pp44

Deemer Pete, Benefield Gabrielle, Larman Craig, Vodde Bas. Información básica de scrum. 2009. Scrum Training Institute. Traducción Antonoli Leo. Disponible en: [http://www.goodagile.com/scrumprimer/scrumprimer\\_es.pdf](http://www.goodagile.com/scrumprimer/scrumprimer_es.pdf)

Dopacio Cristina. *La financiación de Europa en la pequeña y mediana empresa*. Madrid, España. 2004

DSDM. Dynamic System Development Method. Disponible en: [www.dsdm.org](http://www.dsdm.org)

Dyba T. *An empirical investigation of the key factors for success in software process improvement.*, IEEE Transactions on software engineering, vol 31, No. 5, May 2005, pp. 410-424.

Ericsson H., Pender M. *Business modeling with UML*. UK. Wiley Editorial. 2000.

Espinosa Elvia, Pérez Rebeca. Modelos de Calidad Total en las Pequeñas y Medianas Empresas Mexicanas. UAM.1997 Disponible en: <http://www.azc.uam.mx/publicaciones/gestion/num11y12/doc17.htm>

Franch Xavier. Introducción al diseño de bases de datos. Ed. UOC. Pp8

Flores Sergio, Mendoza Nicandro, Román Armando, Rosas Luis, 2011. Modelo de calidad para la microempresa basado en MoproSoft. CiComp 2011. Baja California, México.

Guerrero F., Etevoric Y. *Adopting the SW-CMM in a Small IT Organization*, IEEE Computer Society, 2004, pp. 29.

Goguen, Joseph A. y Linde, Charlotte. Techniques for requirements elicitation. En: THAYER, Richard y DORFAM, Merlin. *Software Requirements Engineering*. 2 ed. Los Alamitos, California: IEEE Computer Science Press, 2000. p. 140-152.

Gómez Juan. Fundamentos de la metodología RUP. Universidad Tecnológica de Pereira. 16/09/2007. <http://www.scribd.com/doc/297224/RUP>

Goodman Sam, *Administración de efectivo para empresas pequeñas y medianas*, 1988 Mc Graw Hill.

Gutiérrez Edna, Gutiérrez Agustín, Aurora Pérez, Luis Márquez. Acerca de la Implementación de los modelos de calidad en la construcción de software en México. *Revista Digital Universitaria*. Vol. 9. No. 9. 2008. Disponible en: <http://www.revista.unam.mx/vol.9/num9/art73/art73.pdf>

Joskowicz José. Reglas y Prácticas en eXtreme Programming 2008. Disponible en : <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>

Kendall K., Kendall J. (2005) *Análisis y diseño de sistemas*. 6ta ed. E.U.

Landeau Rebeca. *Elaboración de trabajo de investigación*. 2007. Venezuela. Pp 122.

Luna Patricia, Chaparro Ricardo. *Propuesta Metodológica de Soluciones Administrativas para Pequeñas y Medianas Empresas de Software*. XVII Congreso Nacional y III Congreso Internacional de Informática y Computación de ANIEI, Octubre del 2004. Tepic Mayarit.



Mejía Pedro. *Administración de Proyectos. Organizar, planear y calendarizar proyectos de Software*. Abril de 2004. Disponible en:  
<http://delta.cs.cinvestav.mx/~pmejia/softeng/cap 3.ppt>.

Mendoza María. *Metodologías de Desarrollo de Software*. 2004. Disponible en:  
[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)

Moliner Francisco. *Informaticos Generalitat Valenciana*. Madrid. 2005. Pp78.

Mon Alicia, Estayno Marcelo, Arancio Andrea. Universidad Nacional de la Matanza, Argentina. *Desarrollo de un Método de Evaluación de la Madurez para pequeñas y medianas empresas de Software*. 2008

Norma NMX-I-059/02-NYCE-2005 (MoproSoft). Parte 02 Requisitos de procesos. 2005. México. ISBN 970-9841-08-4.

Oktaba Hanna. *Software Process Improvement for Small and Medium Enterprises*. EU. 2008. Pp 61-64

Oktaba Hanna. *Modelo de Procesos para la Industria de Software MoProSoft*, versión 1.3. México. 2005

Oktaba2 H. *Tejiendo nuestra red. Investigación de Procesos.*, Revista Software Guru Conocimiento en Práctica, Año 01, No.5, 2005, pp. 6.

Paulk, M. *Extreme Programming from CMM Perspective*. IEEE Software. November/December. 2001.

Peñaloza Marcela. Sugerencias para la elaboración de contratos informáticos para el desarrollo de software. Año 5, Número 45, Enero de 2006. Disponible en:  
<http://www.enterate.unam.mx/Articulos/2006/enero/sugerenc.htm>

Pressman Roger. *Ingeniería del Software Un enfoque Practico*, Mc. Graw Hill. 2005

PROSOFT, *Programa para el Desarrollo de la Industria de Software*. 3er. Trimestre 2005. México.

Raghavan S., Zelesnik G., & Ford G. "Lecture Notes on Requirements Elicitation". Educational Materials CMU/SEI-94-EM-10, Software Engineering Institute, Carnegie Mellon University. 1994  
<http://www.scielo.cl/pdf/ingeniare/v18n1/art04.pdf>

Rainer A., Hall T. *A quantitative and qualitative analysis of factors affecting software processes*. Submitted to the *Journal of Systems and Software*. Department of Computer Science. University of Hertfordshire. 2003. Disponible en:  
<http://portal.acm.org/citation.cfm?id=945041>

Ruiz Francisco, Polo Macario. *Mantenimiento del Software*. Universidad de Castilla-La Mancha. 2001. Disponible en: <http://alarcos.inf-cr.uclm.es/per/fruiz/cur/mso/trans/s3.pdf>

Schwaber K , Beedle M., Martin R.C. *Agile Software Development with SCRUM*. Prentice Hall. 2001

Sommerville Ian. *Procesos de la Ingeniería de requerimientos*. 7ma Edición. Madrid, 2005

Stair R., Reynolds W. (2000) Principios de sistemas de información: enfoque administrativo. 4ed. E.U.

Stapleton, Jennifer. 1997. DSDM: Dynamic Systems Development Method. Harlow, England. Addison-Wesley.

Tuya Javier, Ramos Isabel, Dolado Javier. Técnicas cuantitativas para la gestión en la ingeniería del software. Pp58 2007, España

Valenzuela Leonel, Flores Brenda, Olgúin Martín. Arquitectura para la coordinación de Flujos de Trabajo de MoProSoft por niveles de capacidad de procesos. 1er. Congreso Internacional de Ciencias Computacionales (CICOMP). 2006. México. Pp. 7-12.

Valenzuela Leonel, Flores Brenda. Especificación formal de elementos MoProSoft a partir del modelo de referencia de flujo de trabajo. Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento. 2008. Ecuador. Pp. 197-204.

Vázquez R. PYMES y la vanguardia tecnológica en sistemas de información. 2002.

Disponible en:

<http://www.gestiopolis.com/recursos/documentos/fulldocs/emp/pymesssii.htm>

Vega Carlos, Rivera Laura, García Arturo. Mejores Prácticas para el establecimiento y aseguramiento de la calidad del Software. Universidad de Málaga. 2008. Disponible en:

<http://www.eumed.net/libros/2008a/351/Modelo%20de%20Madurez%20de%20Capacidad.htm>

Wells, Don. 1999. The rules of extreme programming. Disponible en:

<http://www.extremeprogramming.org/rules.html>