



Universidad Autónoma de Querétaro
Facultad de Informática

ANÁLISIS DEL CONJUNTO DE ESTÁNDARES DE
CALIDAD DE SOFTWARE DEL IEEE PARA SU
APLICACIÓN EN LA MIPYME DE DESARROLLO DE
SOFTWARE, EN LA REGIÓN

Tesis

Que como parte de los requisitos para obtener el grado de

Maestro en

INGENIERÍA DE SOFTWARE DISTRIBUIDO

Presenta

CLAUDIA RODRÍGUEZ LEMUS

Santiago de Querétaro, 2009

La presente obra está bajo la licencia:
<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>



CC BY-NC-ND 4.0 DEED

Atribución-NoComercial-SinDerivadas 4.0 Internacional

Usted es libre de:

Compartir — copiar y redistribuir el material en cualquier medio o formato

La licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

Bajo los siguientes términos:



Atribución — Usted debe dar [crédito de manera adecuada](#), brindar un enlace a la licencia, e [indicar si se han realizado cambios](#). Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.



NoComercial — Usted no puede hacer uso del material con [propósitos comerciales](#).



SinDerivadas — Si [remezcla, transforma o crea a partir](#) del material, no podrá distribuir el material modificado.

No hay restricciones adicionales — No puede aplicar términos legales ni [medidas tecnológicas](#) que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

Avisos:

No tiene que cumplir con la licencia para elementos del material en el dominio público o cuando su uso esté permitido por una [excepción o limitación](#) aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como [publicidad, privacidad, o derechos morales](#) pueden limitar la forma en que utilice el material.

Universidad Autónoma de Querétaro
Facultad de Informática
Maestría en Ingeniería de Software Distribuido

ANÁLISIS DEL CONJUNTO DE ESTÁNDARES DE CALIDAD DE SOFTWARE
DEL IEEE PARA SU APLICACIÓN EN LA MIPYME DE DESARROLLO DE
SOFTWARE, EN LA REGIÓN

TESIS

Que como parte de los requisitos para obtener el diploma/grado de (o la)

MAESTRO EN INGENIERÍA DE SOFTWARE DISTRIBUIDO

Presenta:

CLAUDIA RODRÍGUEZ LEMUS

Dirigido por:

M.C. LUIS RODRIGO VALENCIA PÉREZ

SINODALES

M.C. LUIS RODRIGO VALENCIA PÉREZ

Presidente

Firma

M.C. ROSA MARÍA ROMERO GONZÁLEZ

Secretario

Firma

DR. EFRÉN GORROSTIETA HURTADO

Vocal

Firma

DR. JESÚS CARLOS PEDRAZA ORTEGA

Suplente

Firma

M.I. JUAN MANUEL PEÑA AGUILAR

Suplente

Firma

M.C. RUTH ANGÉLICA RICO HERNÁNDEZ

Director de la Facultad

DR. MARCO ANTONIO ACEVES
HERNÁNDEZ

Director de Investigación y
Posgrado

Centro Universitario
Querétaro, Qro.
2009

México

RESUMEN

El tiempo promedio de vida de una miPyme con giro en el desarrollo de software en México es de dos años y medio. Una de las razones que se considera influye en el fracaso de las mismas es la poca o nula importancia que se le da al uso de estándares que garanticen tanto a estas como a sus clientes, un producto con calidad reconocida. El presente trabajo pretende transmitir a estas empresas que la calidad no debe ser tomada como una moda, sino como una forma de vida a la que es imprescindible entender, aprender y adoptar o de lo contrario condenarse a desaparecer. Para lograr lo anterior se planteó como objetivo principal de la investigación el de seleccionar del conjunto de estándares de desarrollo de software de la IEEE, aquellos que permitan su implementación durante las diferentes fases del ciclo de vida del sistema, clasificadas por áreas de conocimiento. Posteriormente, se presenta una propuesta de implantación de los estándares de acuerdo al ciclo de vida en cascada y la operación de acuerdo a proyectos que desarrolle la miPyme en un tiempo determinado. Para llevar a cabo el desarrollo de la propuesta se ha hecho uso de la información que proporcionan los estándares de la IEEE, además del modelo CMM para determinar el nivel de madurez de los procesos empleados de la miPyme; terminando con una recomendación de aplicar la mejora continua para garantizar la calidad aún con una miPyme cambiante y en crecimiento.

(Palabras clave: estándares, miPymes, prosoft)

ÍNDICE GENERAL

ÍNDICE GENERAL	4	
ÍNDICE DE FIGURAS	6	
ÍNDICE DE TABLAS	7	
PARTE I	INTRODUCCIÓN	8
CAPÍTULO I INTRODUCCIÓN	9	
1.1 Antecedentes	9	
1.2 Importancia	11	
1.3 Justificación	13	
CAPÍTULO II FUNDAMENTOS	17	
2.1 Marco Teórico	17	
2.1.1 Las empresas de desarrollo de software	17	
2.1.2 Calidad del Software	22	
2.1.3 Sistema de calidad	26	
2.1.4 Los estándares de calidad de software de la IEEE	29	
2.2 Desarrollo Histórico	33	
2.2.1 Las MiPyMes en México	33	
2.2.2 Una colección integrada de estándares	35	
CAPÍTULO III PLANTEAMIENTO DEL PROBLEMA	40	
3.1 Descripción del Problema	40	
CAPÍTULO IV PLANTEAMIENTO DE LA HIPÓTESIS	42	
4.1 Hipótesis General	42	
CAPÍTULO V OBJETIVOS	43	
5.1 Objetivo General	43	
5.2 Objetivos Específicos	43	
5.3 Metas	43	
PARTE II	PROPUESTA DE METODOLOGÍA	44
CAPÍTULO VI PROPUESTA DE SOLUCIÓN	45	
6.1 Análisis de los 40 estándares de software de la IEEE	45	
6.1.1 Organización de la colección de estándares de la ingeniería de software de la IEEE	46	
6.2 Perfil de las empresas de Software en México	48	
6.3 Propuesta de Aplicación.	49	
CAPÍTULO VII ANÁLISIS Y CLASIFICACIÓN	50	
7.1 Selección de la clasificación	50	
7.2 Descripción de los estándares de calidad de software de la IEEE	50	
7.2.1 Área de conocimiento de Requerimientos de Software.	51	
7.2.2 Área de conocimiento de Diseño de Software.	54	
7.2.3 Área de conocimiento de Construcción de Software.	57	
7.2.4 Área de conocimiento de Prueba de Software.	59	

7.2.5 Área de conocimiento de Mantenimiento de Software.	61
7.2.6 Área de conocimiento de Administración de la Configuración de Software.	62
7.2.7 Área de conocimiento de Administración de la Ingeniería de Software.	64
7.2.8 Área de conocimiento de Procesos de Ingeniería de Software.	66
7.2.9 Área de conocimiento de Herramientas de Ingeniería de Software y Métodos.	69
7.2.10 Área de conocimiento de Calidad de Software.	72
7.2.11 Área de conocimiento de Ingeniería de Sistemas de Software.	75
7.2.12 Área de conocimiento de Mejores Prácticas de Internet	77
7.3 Estructuras Organizacionales.	78
CAPITULO VIII DESARROLLO	80
8.1 Selección del tipo empresa a estudiar	80
8.2 Selección del sector del área de conocimiento a estudiar.	83
8.3 Selección de estándares	84
8.4 Actividades a realizar de acuerdo al ciclo de vida de cascada	84
8.5 Actividades a realizar de acuerdo al tiempo	86
8.5.1 Análisis del estado actual.	86
8.5.2 Implantación del estándar	87
8.5.3 Maduración	91
8.5.4 Mejora Continua	96
PARTE III	RESULTADOS
	101
CAPÍTULO IX RESULTADOS	102
CAPÍTULO X CONCLUSIONES	103
CAPÍTULO XI TRABAJO FUTURO	104
BIBLIOGRAFÍA	105
PARTE IV	ANEXOS
	107

ÍNDICE DE FIGURAS

Fig. no. 1. Informe sobre gastos en proyectos de desarrollo de software según la GAO.	12
Figura no. 2. Actividades, métodos y prácticas utilizadas en la producción y evolución del software.	14
Figura no.3 Modelo de Evolución Empresarial.	19
Figura no. 4. Empresas de software	20
Figura no. 5. Pirámide que muestra la métrica de los procesos del desarrollo del software.	23
Figura 6. Perfil KPA resultado de la evaluación CMM para el aseguramiento de la calidad	24
Figura no. 7. Arquitectura del Comité SES para su colección de estándares.	36
Figura 8. Relación de páginas por documentos de los estándares de software de la IEEE.....	46
Figura no. 9. Proporción de servicios que prestan las empresas	79
Figura no. 10. Organización de estándares de acuerdo al ciclo de vida.....	86
Figura no. 11. Implantación del estándar A por proyecto.	88
Figura no. 12. Propuesta de implementación de los estándares seleccionados por proyecto.	90
Figura no. 13. Fase de verificación de procesos	92
Figura no. 14. Niveles de Madurez en los procesos del Software	93
Figura no. 15. Diagrama de flujo para alcanzar la evaluación y control de los procesos	95
Figura no. 16. Diagrama de flujo para garantizar la mejora continua.....	98
Figura no. 17. Ciclo de la calidad	99

ÍNDICE DE TABLAS

Tabla 1. Factores de éxito de la industria del software en países líderes	18
Tabla 2. Composición y Distribución de las MiPyMEs en México (Sec. Econ de México).	33
Tabla 3. Empresas de TI.	35
Tabla no. 4 Empresas desarrolladoras de software en México	48
Tabla no. 5. Listado de Estándares que corresponden al área de Requerimientos de Software	52
Tabla no. 6. Listado de Estándares que corresponden al área de Diseño de Software.....	55
Tabla no. 7. Listado de Estándares que corresponden al área de Construcción de Software	58
Tabla no. 8. Listado de Estándares que corresponden al área de Pruebas del Software	60
Tabla no. 9. Listado de Estándares que corresponden al área de Mantenimiento de Software	61
Tabla no. 10. Estándares correspondientes al área de Admón. de la Configuración del Software	63
Tabla no. 11. Estándares correspondientes al área de Administración de la Ingeniería de Software ...	64
Tabla no. 12. Listado de Estándares que corresponden al área de Procesos de Ingeniería de Software	67
Tabla no. 13. Estándares correspondientes al área de Herram. de Ingeniería de Software y Métodos.	70
Tabla no. 14. Listado de Estándares que corresponden al área de Calidad de Software.....	73
Tabla no. 15. Listado de Estándares que corresponden al área de Ingeniería de Sistemas de Software	76
Tabla no. 16. Listado de Estándares que corresponden al área de Mejores Prácticas de Internet	78
Tabla no. 17. Selección de estándares.	84
Tabla no. 18. Áreas de proceso claves para CMM	94

PARTE I

INTRODUCCIÓN

CAPÍTULO I INTRODUCCIÓN

1.1 Antecedentes

La presente tesis contiene los resultados de una investigación sobre el conjunto de los cuarenta estándares de calidad de software establecidos por la IEEE, con la finalidad de servir a las miPyMes de software y a las áreas de desarrollo de software como una guía para determinar los estándares que se adecuen a ellas.

En el año de 1994, en una empresa de transporte de carga, se estaba realizando un sistema de nómina, facturación y control de inventarios; el cual estaba siendo desarrollado por un Ingeniero Químico desde hacía más de tres años. La inquietud del sistema nació cuando el dueño de la empresa conoció un paquete que le ayudaría a llevar sus procesos, pero que costaba más de \$100,000.00 (cie mil pesos). El ingeniero le prometió que él se lo desarrollaría por la cuarta parte y enfocado a su problemática. Sin embargo, en todo ese tiempo, el sistema no podía ser terminado y el Ingeniero ya había cobrado más de lo que le hubiera costado el paquete al dueño de la empresa.

La situación anterior, como muchas otras; se debió a que el desarrollo de software en la región Bajío del País, al principio carecía de métodos que ayudaran a medir su efectividad y el tiempo real que este requería, además de la ingeniería de software necesaria para cubrir los verdaderos requerimientos del usuario; ya que muchos de los primeros desarrolladores, ni siquiera tenían el perfil de programador. Esto ocasionó en muchos casos, el desarrollo eterno de software; es decir: Aquél software que siempre estuvo en proceso de desarrollo y que solo se llegó a utilizar una parte de él o en el peor de los casos, nunca se pudo usar; y que sin embargo estuvo siendo pagado por el usuario todo ese tiempo.

En los casos en los que si se logró terminar el desarrollo, en muchos de ellos, hubo fallas en la etapa de análisis, lo que provocó que el software resultara no ser la solución por la cual se había iniciado su desarrollo. En otros, el software no abarcaba la problemática completa y en la mayoría, el software se entregó al menos 6 meses después de lo previsto en un inicio.

Lo anterior aunado a los problemas de comunicación entre el usuario y los desarrolladores, provocaron la desconfianza y desinterés de los usuarios por adquirir “software a la medida” y preferir comprar software comercial, que quizás no solucionaría todos sus problemas, pero al menos, sería seguro de tener.

Hoy en día existen miPyMes (Micro, pequeñas y medianas empresas) en la región bajo integradas por programadores expertos, que se dedican a desarrollar software, y que están buscando un lugar en ese mercado. Sin embargo, continúan con problemas de planeación en sus procesos en cuanto a tiempos de entrega, cálculos de costos de desarrollo y de levantamiento de requerimientos.

Si es difícil para las miPyMes de desarrollo de software vender sus productos en el País, lo es más difícil si se piensa en exportar, ya que se requiere de la utilización de estándares que garanticen la calidad del software, y que ayude a resolver las problemáticas a las que se enfrentan.

Sin embargo, todavía estas miPyMes no están convencidas de que el uso de estándares de calidad les lleve a acortar sus tiempos de desarrollo, ya que piensan que se pierde más tiempo llenando formatos.

Lo descrito anteriormente fue el principal interés en desarrollar esta investigación; ya que la presente tesis proveerá a las miPyMes de desarrollo de software una guía para utilizar los estándares de calidad en sus procesos, así como de mostrarles que son herramientas en las que se pueden apoyar para

asegurar la calidad de sus productos; y a su vez, garantizar a los usuarios la entrega de un software eficiente y en los tiempos establecidos; esto provocará que las miPymes sean más competitivas e incluso, tengan seguro un lugar en el mercado internacional.

1.2 Importancia

El hecho de que las miPyMes no cuenten con estándares de calidad, implica los siguientes problemas:

- a) En el proceso de desarrollo, desviar los objetivos iniciales del sistema.
Un error muy común en el programador actual, es el no documentar las etapas de vida del sistema, de tal forma que corre el riesgo de perder el rumbo en alguna etapa y esto repercute en el sistema final, ya que muchas veces el resultado no resolvió la problemática por la cual fue desarrollado.

- b) Mala planeación:
 - b.1) El programador no cuenta con herramientas de medición para determinar los tiempos, la cantidad y el tipo de recursos humanos, de cada una de las etapas: Según [Robles, 2003] En el año de 1998, los Estados Unidos realizaron una investigación que indicó que se había invertido 3, 661 mil millones de pesos en proyectos de software, de los cuales, solo el 26% cumplió tanto los presupuestos, como su planificación. Además, las pérdidas debidas a fallos y retrasos fueron de 1,292.95 mil millones de pesos. En total, las pérdidas globales fueron de 7,485.5 mil millones de pesos.

Otras estadísticas de gasto de proyectos de software se encuentran en [Barrios, 2006], en donde se indica que según la Oficina de

Contabilidad del gobierno de Estados Unidos (Government Accounting Office), en el año de 1979, el 47% del software desarrollado, nunca se utilizó (ver figura no. 1).

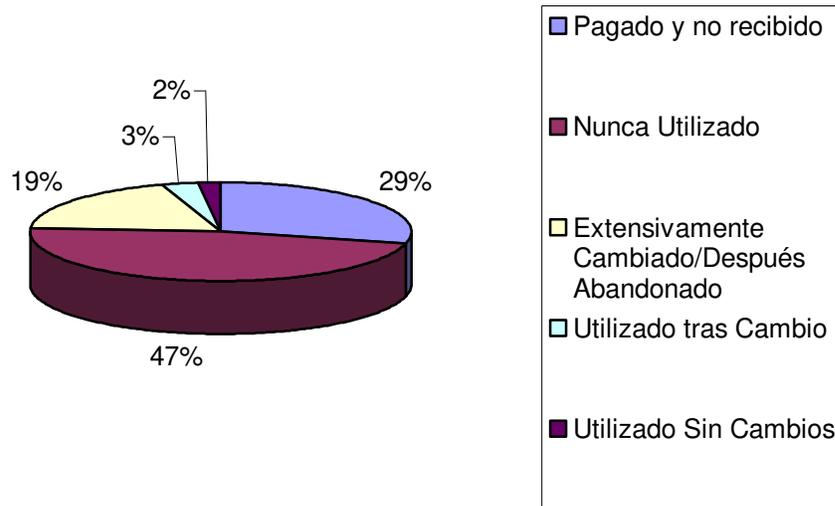


Fig. no. 1. Informe sobre gastos en proyectos de desarrollo de software según la GAO.

b.2) Mala elección del lenguaje de programación o de la plataforma:

Debido a la mercadotecnia actual, los programadores que no aplican estándares de calidad, no realizan un análisis efectivo sobre el lenguaje de programación y plataforma adecuada para sus clientes, lo que conlleva a problemas de compatibilidad al momento de la ejecución.

b.3) Falta de herramientas para la determinación de los requerimientos:

En muchas ocasiones el cliente de la empresa debe hacer inversiones de equipo o software no contempladas desde un inicio.

c) Pérdida de oportunidades de trabajo:

Actualmente existe una oferta muy grande de desarrolladores que no es directamente proporcional a la demanda. Esto obliga a los desarrolladores a buscar elementos como las certificaciones, que le

permitan participar en la competencia, asegurando una calidad total en sus servicios a sus clientes.

Debido a estas problemáticas, se ha dejado de creer en el software a la medida y se ha optado por adquirir software comercial. Esto asegura al cliente su adquisición, pero no la cobertura de sus necesidades.

Por lo anterior, se espera con este proyecto, un acercamiento de la micro, pequeña y mediana empresa de software de México a los estándares de calidad de software, específicamente de la IEEE, los cuales le permitirán a estas empresas entregar sistemas con calidad a sus clientes y a su vez estos tendrán la confianza nuevamente de comprar software a la medida.

1.3 Justificación

Ya anteriormente se mencionaron las problemáticas que las miPyMes han presentado a lo largo de la historia en los diferentes procesos que utilizan para el desarrollo de sus productos. Para continuar con el desarrollo de este trabajo, se requirió de este apartado que muestra la justificación de la investigación.

Debido a la globalización, hoy en día las empresas de software que aseguran su éxito son aquellas que utilizan estándares de calidad en sus procesos para fabricar software de calidad (ver figura no.2).

La obtención de un software con calidad según [Fernández, 1995], implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

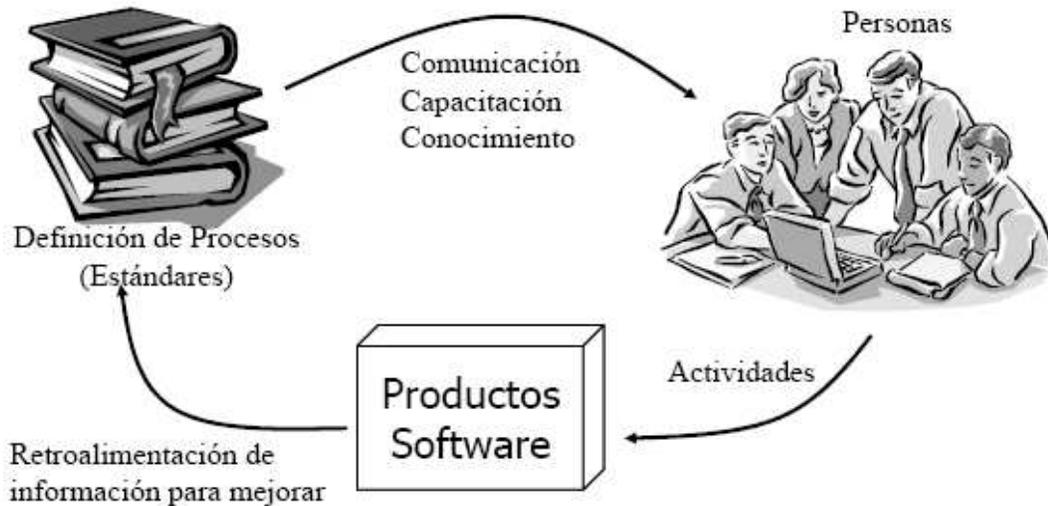


Figura no. 2. Actividades, métodos y prácticas utilizadas en la producción y evolución del software.

La política establecida debe estar sustentada sobre tres principios básicos: tecnológico, administrativo y ergonómico.

- a) El principio tecnológico define las técnicas a utilizar en el proceso de desarrollo del software.
- b) El principio administrativo contempla las funciones de planificación y control del desarrollo del software, así como la organización del ambiente o centro de ingeniería de software.
- c) El principio ergonómico define la interfaz entre el usuario y el ambiente automatizado.

La adopción de una buena política contribuye en gran medida a lograr la calidad del software, pero no la asegura. Para el aseguramiento de la calidad es necesario su control o evaluación.

Según [Fernández, 2001] el software es inmaterial y la calidad del software es difícil de medir, pero tenemos algunas pautas, algunos indicadores que nos ayudan a diferenciar los productos de calidad de los que carecen de ella:

- El acercamiento a cero defectos.

- El cumplimiento de los requisitos intrínsecos y expresos.
- La satisfacción del cliente

Sobre todo la satisfacción del cliente. Ya se sabe que un suministrador puede engañar a todos alguna vez o a alguno muchas veces, pero no puede engañar a muchos durante largo tiempo. El cliente “casi” siempre tiene razón y para eso están las encuestas de satisfacción. El software de calidad es el que resulta en los primeros puestos de la tabla por “aclamación” de los usuarios.

El argumento de la calidad es exhibido por las empresas como un factor diferenciador, como clave de sus procesos de negocio y como eslogan de competitividad empresarial. De hecho, la exigencia cada vez mayor por parte del mercado de la certificación ISO 9000 o el interés creciente por los modelos de calidad de gestión empresarial de tipo EFQM) son indicadores de la percepción de la calidad como un elemento cada vez más necesario.

La calidad del software debe ser una disciplina más dentro de la Ingeniería del software. El principal instrumento para garantizar la calidad de las aplicaciones sigue siendo el Plan de Calidad. El plan se basa en unas normas o estándares genéricos y en unos procedimientos particulares.

Los procedimientos pueden variar en cada organización, pero lo importante es que estén escritos, personalizados, adaptados a los procesos de la organización y, lo que es más importante, que se cumplan. La calidad del software debe implementarse a lo largo de todo el ciclo de vida, debe correr paralela desde la planificación del producto hasta la fase de producción del mismo. Para ello se cuenta con una serie de ayudas, a las que el grupo dedica su atención y trabajo, a través de distintas actividades:

- Para la fase de Planificación se pueden utilizar elementos y herramientas propias de la Gestión de proyectos.
- Para los procesos de Análisis y Diseño deberemos contar con herramientas de análisis y de métricas, entre otras.
- En los procesos de Construcción y pruebas deberíamos echar mano de herramientas de configuración y simulación
- Y finalmente, para el proceso de Producción, básicamente habremos de utilizar herramientas de monitorización y pruebas de producción.

Se puede concluir que las miPyMes que utilicen estándares de calidad para sus procesos, tendrán mayores oportunidades en el mercado nacional e incluso mundial. Pero el hecho de que sean tantos estándares de calidad para cada proceso, y si se toma en cuenta además de que no todos esos estándares se pueden aplicar a todas las empresas (principalmente por el tamaño de la empresa), puede provocar que estas empresas caigan en el desánimo y terminen por no utilizarlos. Es por eso la importancia de esta investigación, que permitirá a esas empresas, consultar cuáles estándares son los más adecuados según sus características, y que le sirvan además para entrar en el mercado con calidad.

CAPÍTULO II FUNDAMENTOS

2.1 Marco Teórico

A continuación se presenta una investigación acerca de los temas más importantes que se requirieron para el desarrollo de esta tesis; en donde se mostrará información sobre las miPyMes de software, la ingeniería del software y sobre la calidad del software.

2.1.1 Las empresas de desarrollo de software

La siguiente información, proviene del proyecto del gobierno mexicano para impulsar a la industria del software, apoyándose de los estándares de calidad.

2.1.1.1 Marco Mundial

El mercado de TIC representa el 6.6% del valor de la producción económica mundial. Durante la década de los noventa la mayor parte de los países, aún los que enfrentaron crisis financieras y recesiones económicas, incrementaron su gasto en tecnologías de información y comunicación.

En los años recientes, el gasto en tecnologías de información ha trasladado su énfasis del hardware al software, provocando que la relación entre el segundo y el primero suba de 32.5% en 1996 a 40% en 1999. El mercado mundial de productos de software rebasa los 153,000 millones de dólares anuales. Estados Unidos es el principal consumidor, con un gasto superior a los 75,000 millones de dólares anuales y una participación de 48.8% en el total mundial [e-Economía, 2004].

Si bien, los países desarrollados continúan siendo líderes en esta materia, la demanda creciente no puede ser satisfecha con su oferta interna. Por ejemplo, en Estados Unidos existe una brecha entre oferta y demanda de más de 800,000 desarrolladores de software que, simplemente por el tamaño relativamente reducido de la población joven de ese país, no podrá eliminarse antes del año 2010 [e-Economía, 2004].

Una proporción creciente de la producción mundial de software se realiza en países en desarrollo. La India e Irlanda representan dos casos exitosos de creación y crecimiento de industrias nacionales basadas en la exportación.

Tabla 1. Factores de éxito de la industria del software en países líderes [e-Economía, 2004]

Concepto/País	India	Irlanda	Estados Unidos Y Canadá	Israel	Singapur
Aseguramiento de la calidad	X				
Costo bajo de recursos y operación	X			X	X
Diferenciamiento de Recursos Humanos	X				
Maquila para empresas de otros países (offshoring)	X				
Outsourcing	X				
Política gubernamental	X				
Recursos humanos	X				
Alianzas internacionales		X	X		
Competencia de empresas		X			
Competencias administrativas de globalización de recursos humanos y del control de proyectos. Certificación administrativa y de estrategias de negocios.		X	X		
Entrepreneurship					
Desarrollo de industrias relacionadas (infraestructura de apoyo financiero, físico ambiental, fiscal y logístico ofrecidos por el gobierno y la IP)		X	X		
Diferenciamiento de recurso humano (niveles profesionales o grado de especialización)		X			
Estabilidad política y económica		X	X		
Conocimiento del idioma Inglés		X	X		
Infraestructura en telecomunicaciones		X	X		X
Niveles educativos elevados		X			
Desarrollo de la industria doméstica			X		
Experiencia acumulada			X		
Innovación tecnológica			X		
Marco normativo (garantía de aplicación de los derechos de propiedad intelectual e industrial)			X		

Nichos especializados de mercado	X		
Competencia ingenieril y tecnológica		X	
Inversión extranjera			X
Aprovechamiento de diferencias de horario respecto de Estados Unidos para un trabajo ligado y continuo			X

2.1.1.2 Entorno productivo nacional e industria del software

La relación que hay entre los niveles de competitividad de las empresas que constituyen el mercado objetivo de las empresas de software y éstas, se caracteriza por una influencia mutua. Por un lado, el uso adecuado de tecnologías de información genera mejoras en la productividad; por el otro, las empresas con mejor evolución empresarial son las mayores consumidoras de esas tecnologías. Por ello, es conveniente analizar las etapas de evolución de las empresas, de acuerdo con los niveles de modernización tecnológica, que determinan su competitividad. Un modelo de evolución empresarial basado en el nivel de competitividad del entorno productivo se muestra en la figura 3 [e-Economía,2004]

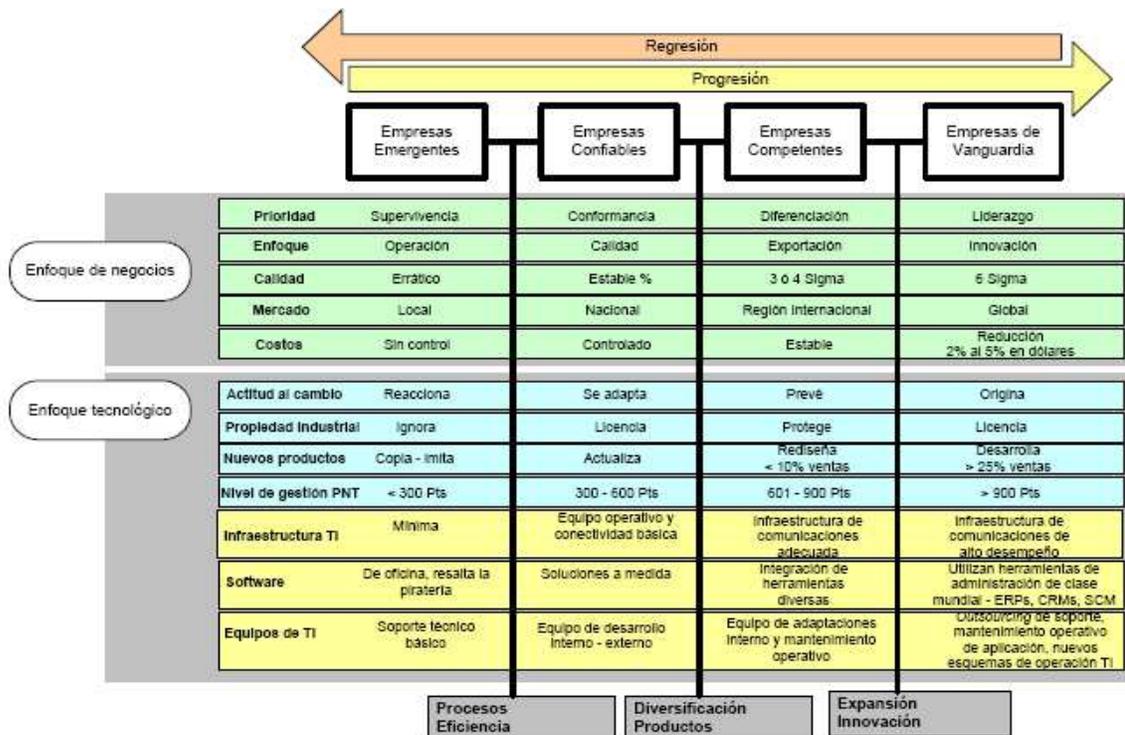


Figura no.3 Modelo de Evolución Empresarial.

En este modelo se observa en las últimas tres categorías los elementos relacionados con las tecnologías de información (TI). Las etapas que catalizan cambios en las categorías de competitividad son: eficiencia de procesos, diversificación de productos, expansión e innovación.

En nuestro país, el 98% de las empresas, que son el mercado objetivo local de las empresas de software mexicanas, son de carácter emergente en términos de competitividad; esto significa que la gran mayoría de las unidades productivas consumen poco software y de baja sofisticación.

La misma categorización aplicada a las empresas de software da el resultado que se muestra en la figura no. 4 [e-Economía, 2004]].

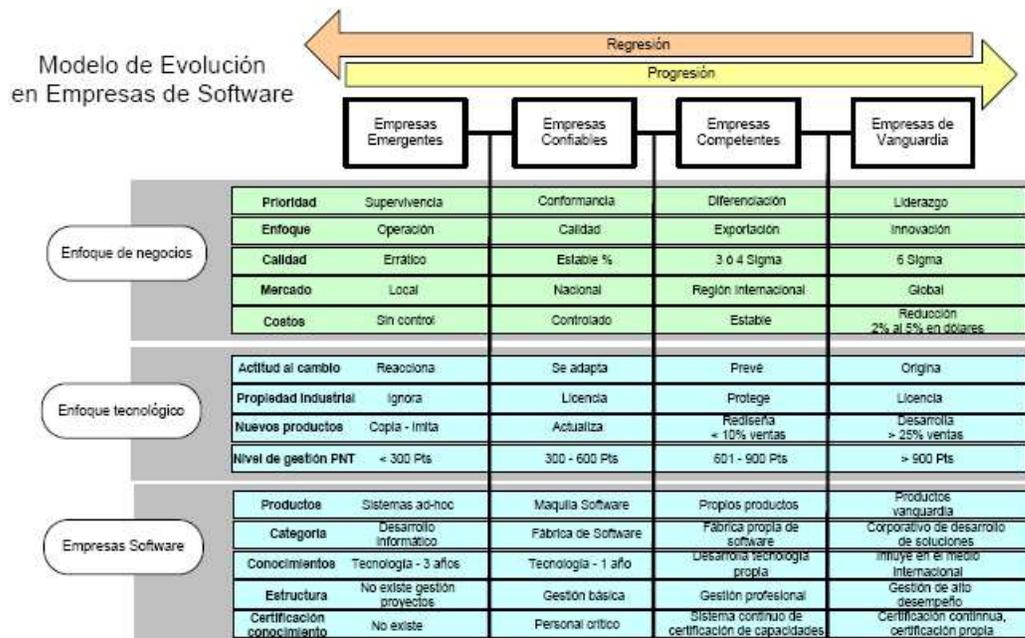


Figura no. 4. Empresas de software

2.1.1.3 Prosoft

El Fondo del Programa para el Desarrollo de la industria del Software (PROSOFT) tiene como objetivo impulsar el desarrollo de la industria de las tecnologías de información a través del otorgamiento de subsidios de carácter temporal a proyectos que estén dirigidos a la creación, desarrollo, consolidación, viabilidad, productividad, competitividad y sustentabilidad de las empresas del sector de Tecnologías de Información y servicios relacionados.

2.1.1.4 Población Objetivo

Las personas físicas con actividad empresarial o las personas morales del sector de TI; los organismos, grupos de empresas, empresas integradoras y asociaciones civiles sin fines de lucro del sector de TI; las instituciones académicas y los emprendedores de este sector económico así como los organismos públicos, privados o mixtos sin fines de lucro entre cuyos objetivos se encuentre el desarrollo del sector de TI.

Se consideran empresas del sector de TI y por ende de la industria de software y servicios relacionados, las que realizan como actividad económica alguna de las siguientes:

- Desarrollo de software empaquetado;
- Desarrollo de software de sistema y herramientas para desarrollo de software aplicativo;
- Desarrollo de software aplicativo;
- Servicios de consultoría de software;
- Servicios de mantenimiento y soporte de sistemas computacionales;
- Servicios de análisis de sistemas computacionales;
- Servicios de diseño de sistemas computacionales;
- Servicios de programación de sistemas computacionales;

- Servicios de procesamiento de datos;
- Servicios de diseño, desarrollo y administración de bases de datos;
- Servicios de implantación y pruebas de sistemas computacionales;
- Servicios de integración de sistemas computacionales;
- Servicios de mantenimiento de sistemas computacionales y procesamiento de datos;
- Servicios de seguridad de sistemas computacionales y procesamiento de datos;
- Servicios de análisis y gestión de riesgos de sistemas computacionales y procesamiento de datos;
- Procesos de negocio relacionados con sistemas computacionales y comunicaciones;
- Servicios de valor agregado de análisis, diseño, desarrollo, administración, mantenimiento, pruebas, seguridad, implantación, mantenimiento y soporte de sistemas computacionales, procesamiento de datos y procesos de negocio;
- Servicios de capacitación, consultoría y evaluación para el mejoramiento de la capacidad humana, aseguramiento de la calidad y de procesos de las empresas del sector de TI;

2.1.2 Calidad del Software

La calidad del software, según [Fernández, 1995] es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

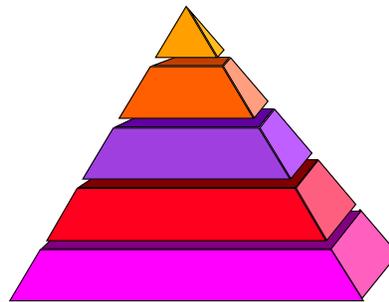
Además, los escritores agregan que la calidad del software es medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable al nivel de "cero fallas"; un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad; mientras que un producto de software para ser explotado durante un largo

período (10 años o más), necesita ser confiable, mantenible y flexible para disminuir los costos de mantenimiento y perfeccionamiento durante el tiempo de explotación.

La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software ver figura no. 5 [Spawar, 2004].

Reacción para mejorar los procesos y la productividad

Métrica de los Procesos



Métrica de los Proyectos
Rastree el progreso del proyecto

Métrica de lo Productos
Rastree la calidad del producto

Figura no. 5. Pirámide que muestra la métrica de los procesos del desarrollo del software.

2.1.2.1 ¿Cómo obtener un software de calidad?

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que

eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software [Fernández, 1995].

La política establecida debe estar sustentada sobre tres principios básicos: tecnológico, administrativo y ergonómico.

El principio tecnológico define las técnicas a utilizar en el proceso de desarrollo del software. El principio administrativo contempla las funciones de planificación y control del desarrollo del software, así como la organización del ambiente o centro de ingeniería de software.

El principio ergonómico define la interfaz entre el usuario y el ambiente automatizado.

La adopción de una buena política contribuye en gran medida a lograr la calidad del software, pero no la asegura. Para el aseguramiento de la calidad es necesario su control o evaluación (ver figura no. 6) [Jenkins, 2003].



Figura 6. Perfil KPA resultado de la evaluación CMM para el aseguramiento de la calidad

2.1.2.2 ¿Como controlar la calidad del software?

Para controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición, ya que, como bien plantea Tom De Marco, "usted no puede controlar lo que no se puede medir" [Fernández et al García y Beltrán, 1995].

Las cualidades para medir la calidad del software son definidas por innumerables autores, los cuales las denominan y agrupan de formas diferentes. Por ejemplo, John Wiley define métricas de calidad y criterios, donde cada métrica se obtiene a partir de combinaciones de los diferentes criterios. La Metodología para la evaluación de la calidad de los medios de programas de la CIC, de Rusia, define indicadores de calidad estructurados en cuatro niveles jerárquicos: factor, criterio, métrica, elemento de evaluación, donde cada nivel inferior contiene los indicadores que conforman el nivel precedente. Otros autores identifican la calidad con el nivel de complejidad del software y definen dos categorías de métricas: de complejidad de programa o código, y de complejidad de sistema o estructura.

Todos los autores coinciden en que el software posee determinados índices medibles que son las bases para la calidad, el control y el perfeccionamiento de la productividad.

Una vez seleccionados los índices de calidad, se debe establecer el proceso de control, que requiere los siguientes pasos:

- Definir el software que va a ser controlado: clasificación por tipo, esfera de aplicación, complejidad, etc., de acuerdo con los estándares establecidos para el desarrollo del software.
- Seleccionar una medida que pueda ser aplicada al objeto de control. Para cada clase de software es necesario definir los indicadores y sus magnitudes.

- Crear o determinar los métodos de valoración de los indicadores: métodos manuales como cuestionarios o encuestas estándares para la medición de criterios periciales y herramientas automatizadas para medir los criterios de cálculo.
- Definir las regulaciones organizativas para realizar el control: quiénes participan en el control de la calidad, cuándo se realiza, qué documentos deben ser revisados y elaborados, etc.

2.1.3 Sistema de calidad

Según [Cueva, 1999], un sistema de calidad es una estructura organizativa, procedimientos, procesos y recursos necesarios para implantar la gestión de calidad.

- El sistema de calidad se debe adecuar a los objetivos de calidad de la empresa
- La dirección de la empresa es la responsable de fijar la política de calidad y las decisiones relativas a iniciar, desarrollar, implantar y actualizar el sistema de calidad.
- Un sistema de calidad consta de varias partes
 - Documentación:
 - ✓ Manual de calidad. Es el documento principal para establecer e implantar un sistema de calidad. Puede haber manuales a nivel de empresa, departamento, producto, específicos (compras, proyectos,...)
 - Parte física: locales, herramientas ordenadores, etc.
 - Aspectos humanos:
 - ✓ Formación de personal
 - ✓ Creación y coordinación de equipos de trabajo
 - Normativas
 - ✓ IEEE
 - ❖ IEEE Std 730-1998, Estándar IEEE para planes de aseguramiento de la calidad del software.

- ❖ IEEE Std 730.1-1995, Guía IEEE para planificar el aseguramiento de la calidad del software
- ✓ ISO
 - ❖ ISO 9000: Gestión y aseguramiento de calidad (conceptos y directrices generales)
 - ❖ Recomendaciones externas para aseguramiento de la calidad (ISO 9001, ISO 9002, ISO 9003)
 - ❖ Recomendaciones internas para aseguramiento de la calidad (ISO 9004)
- ✓ MALCOM BALDRIGE NATIONAL QUALITY AWARD
- ✓ Software Engineering Institute (SEI) Capability Maturity Model (CMM) for software

2.1.3.1 Certificación de Calidad.

Un sistema de certificación de calidad permite una valoración independiente que debe demostrar que la organización es capaz de desarrollar productos y servicios de calidad.

Los pilares básicos de la certificación de calidad son tres [Sanders, 1994] :

- Una metodología adecuada
- Un medio de valoración de la metodología
- La metodología utilizada y el medio de valoración de la metodología deben estar reconocidos ampliamente por la industria

2.1.3.2 Factores que determinan la calidad del software

[Cueva, 1999] clasifica estos factores en tres grupos:

- Operaciones del producto: características operativas
- Revisión del producto: capacidad para soportar cambios
- Transición del producto: adaptabilidad a nuevos entornos

2.1.3.3 La documentación:

Según [Reyes, 2005], en una lista de factores de calidad del software, uno podría esperar encontrar la presencia de una buena documentación como uno de los requisitos. Pero éste no es un factor de calidad separado; la necesidad de documentación es una consecuencia de otros factores de calidad vistos en el acápite anterior. Se pueden distinguir tres tipos de documentación:

- Documentación externa.
- Documentación interna,
- Documentación de la interfaz de un módulo

2.1.3.4 Definición de roles:

Comprador:

Una organización que adquiere un sistema, producto o servicio de software de un vendedor.

Vendedor:

Una organización que participa en un contrato en conjunto con el comprador para el suministro de un sistema, producto o servicio de software, bajo los términos del contrato.

Operador:

La organización que opera el sistema.

Desarrollador:

Una organización que desempeña las actividades de desarrollo (incluyendo análisis de requerimientos, diseño, pruebas de aceptación) durante los procesos del ciclo de vida del software.

Soporte:

Una organización que desempeña las actividades de mantenimiento.

2.1.4 Los estándares de calidad de software de la IEEE

En el presente apartado, se listan y se describen los 40 estándares de calidad de software y su clasificación comprendidos en la IEEE, según [Moore, 1995]:

2.1.4.1 Estándares de terminologías y personalizados

- IEEE Std 610.12-1990, Glosario estándar de la IEEE de terminología de Ingeniería de Software.
- IEEE Computer Dictionary Project 610 (P610) – Diccionario estándar de computación el cual consolida los estándares actuales basados en especialidades individuales. Este estándar reemplaza a los glosarios existentes (e.g., 610.12-1990). Intenta crear un vocabulario común para profesionales del software.
- IEEE Std 1062, 1998 Edition, Prácticas recomendadas por la IEEE para adquisición de software.
- IEEE Std 1220-1998, Estándar IEEE para la aplicación y administración de los procesos de ingeniería de software.
- IEEE Std 1228-1994, Estándar IEEE para planeación de seguridad de software.

- IEEE Std 1233, 1998 Edition, Guía IEEE para desarrollar especificaciones de requerimientos de sistemas.
 - IEEE Std 1362-1998, Guía IEEE para conceptos de sistemas de tecnología de información de documentos de operación.
 - ISO/IEC 12207:1995 -- IEEE/EIA 12207 Conjunto de estándares EU que implementa el estándar ISO/IEC.
- El Estándar EU contiene las siguientes partes:
 - ✓ IEEE/EIA Std 12207.0-1996, Procesos del ciclo de vida del software.
 - ✓ IEEE/EIA Std 12207.1-1997, Datos de los procesos del ciclo de vida del software
 - ✓ IEEE/EIA Std 12207.2-1997, Consideraciones de implementación de los procesos del ciclo de vida del software.
 - Estos estándares reemplazan al estándar MIL-STD-498 (y estándares anteriores). Este estándar militar fue presentado a la IEEE y a la EIA como un estándar EU de la industria y su propósito es ser un (internacional) estándar. Satisface al estándar MIL-Q-9858A (Requerimientos de programas de calidad) y al estándar ISO9000 (Sistemas de calidad) para software.

2.1.4.2 Estándares de Procesos

- IEEE Std 730-1998, Estándar IEEE para planes de aseguramiento de la calidad del software.
- IEEE Std 730.1-1995, Guía IEEE para planificar el aseguramiento de la calidad del software.
- IEEE Std 828-1998, Estándar IEEE para planear la administración de la configuración del software.
- IEEE Std 1008-1987 (Reaff 1993), Estándar IEEE para probar unidades de software.

- IEEE Std 1012-1998, Estándar IEEE Standard para verificación y validación de software.
- IEEE Std 1012a-1998, Suplemento del estándar IEEE para verificación y validación de software: Contiene un mapa al estándar IEEE/EIA 12207.1-1997
- IEEE Std 1028-1997, Estándar IEEE para revisiones de software.
- IEEE Std 1042-1987 (Reaff 1993), Guía IEEE para administración de la configuración del software.
- IEEE Std 1045-1992, Estándar IEEE para métricas de productividad del software.
- IEEE Std 1058-1998, Estándar IEEE para planear administración de proyectos de software.
- IEEE Std 1059-1993, Guía IEEE para planear verificaciones y validaciones de software.
- IEEE Std 1074-1997, Estándar IEEE para desarrollar los procesos del ciclo de vida del software.
- IEEE Std 1219-1998, Estándar IEEE para mantenimiento del software.
- IEEE Std 1490-1998, Guía-Adopción IEEE del estándar PMI. Una guía para el cuerpo de conocimiento de la administración del proyecto. Product Standards
- IEEE Std 982.1-1988, Estándar IEEE diccionario de medidas para producir software confiable.
- IEEE Std 982.2-1988, Guía IEEE para el uso del diccionario estándar de mediciones para producir software confiable.
- IEEE Std 1061-1998, Estándar IEEE para metodologías de métricas de calidad de software.
- IEEE Std 1063-1987 (Reaff 1993), Estándar IEEE para documentación para el usuario del software.
- IEEE Std 1465-1998, Estándar IEEE adopción del estándar internacional ISO/IEC 12199:1994 (E) –Tecnología de information –Paquetes de software –Requerimientos de calidad y prueba.

2.1.4.3 Estándares de recursos y técnicas.

- IEEE Std 829-1998, Estándar IEEE para documentación de pruebas al software.
- IEEE Std 830-1998, Prácticas recomendadas IEEE para documentación de requerimientos del software.
- IEEE Std 1016-1998, Prácticas recomendadas IEEE para descripciones de diseño de software.
- IEEE Std 1044-1993, Estándar IEEE para clasificación de anomalías del software.
- IEEE Std 1044.1-1995, Guía IEEE para clasificación de anomalías del Software.
- IEEE Std 1320.1-1998, Estándar IEEE para el modelo funcional del lenguaje –Sintaxis y semántica para IDEF0
- IEEE Std 1320.2-1998, Estándar IEEE para el modelo conceptual del lenguaje –Sintaxis y semántica para IDEF1X₉₇(IDEF_{object})
- IEEE Std 1348-1995, Práctica recomendada IEEE para la adopción de computadoras y herramientas de ingeniería de software auxiliares (CASE).
- IEEE Std 1420.1-1995, Estándar IEEE para tecnologías de información - -software reusable—modelo de datos para la interoperabilidad de bibliotecas reusables: Modelo de datos de interoperabilidad básica (BIDM).
- IEEE Std 1420.1a-1996, Suplemento al estándar IEEE para tecnologías de información --Software reusable—Modelo de datos para Interoperabilidad de bibliotecas reusables: Marco de certificación activo.
- IEEE Std 1430-1996, Guía IEEE para tecnologías de información— Software reusable—conceptos de operación para la interoperabilidad de bibliotecas reusables.
- IEEE Std 1462-1998, Adopción del estándar IEEE de ISO/IEC 14102:1995—Tecnologías de información-- directrices para la evaluación y selección de herramientas CASE.

2.2 Desarrollo Histórico

A continuación se presentan algunos artículos de varios autores que dieron soporte al presente proyecto.

2.2.1 Las MiPyMes en México

Según [Treviño, 2005], las MiPyMEs en México tienen el gran reto de adecuarse a un mundo globalizado y rápido en sus cambios y demandas. Ya no se deben conformar con los mercados internos, ya no es más su ventaja comparativa. Para poder mantenerse y crecer requieren de apoyarse en herramientas de Tecnologías de Información (TI) como los Sistemas de Soporte a la Decisión (SSD).

El objetivo del artículo es el de mostrar la importancia de los sistemas de soporte de toma de decisiones (SSD) para las MiPyMEs en México.

Las tendencias mundiales actuales en el campo empresarial reconocen en las MiPyMEs una parte fundamental de la economía de cualquier nación. Más del 98% del universo de empresas formales e informales en los distintos países se ubican en este grupo, participando también de forma muy elevada en las ventas totales, las exportaciones, el PIB y el empleo en sus países.

De acuerdo con la Secretaría de Economía de México, el total de MiPyMEs asciende a 531,222 empresas, las cuales representan el 97% del total de las empresas en el país, como se muestra en la tabla 2.

Tabla 2. Composición y Distribución de las MiPyMEs en México (Sec. Econ de México).

Tipo de Empresa	Total	Porcentaje
Micro	497,008	90.79%
Pequeña	34,214	6.25%
Mediana	10,645	1.94%
Grande	5,535	1.01%
Total	547,402	100%

Por su gran volumen, es indudable la importancia de las MiPyMEs en la innovación y su rol en el enfrentamiento de los retos de la competitividad de cada país, dado que estas organizaciones cubren un alto porcentaje del empleo. Las MiPyMEs en México emplean al 78% de la población económicamente activa, y que aportan el 68% del PIB de acuerdo con la Secretaría de Economía.

Hoy en día las empresas operan en un medio ambiente que va cambiando constantemente y esta tendencia se va incrementando, por lo que algunas de las razones por las cuales las decisiones son más difíciles de tomar que en el pasado son:

- Mayor número de alternativas disponibles.
- Mayor costo en los errores administrativos por la complejidad y la magnitud de las.
- Difícil acceso a la información necesaria para tomar decisiones.
- Rapidez con la que se deben de tomar las decisiones.

En México las grandes compañías cuentan con tecnologías que los ayuden a tomar estas decisiones. Pero en las MiPyMEs no es así, aún y que son un factor muy importante en el desarrollo del País. En las MiPyMEs la toma de decisiones muchas veces se lleva a cabo por sentido común, o por lo que piensan los administradores, pero no se lleva un control o procedimiento para poder tomar esas decisiones.

Como resultado de esta investigación el autor encontró que debido a su potencial crecimiento y número de éstas, las MiPyMEs son un nicho que ya algunas empresas de TI están considerando atacar. Compañías como Oracle, Microsoft, Avaya y Nortel son algunas de las que cuentan dentro de sus estrategias de negocio maximizar ventas en ese amplio mercado.

En la tabla 3 se muestran algunas de las grandes empresas de TI que en los últimos tiempos han enfocado parte de sus esfuerzos a desarrollos de proyectos enfocados a las MiPyMEs.

Tabla 3. Empresas de TI.

Empresa	Soluciones
Microsoft	Outlook, Excel, Word, CRM
IBM	PC's, bases de datos
SAP	ERP, CRM
Oracle	ERP, bases de datos
JDEdwards	ERP
Sun Microsystems	Web

2.2.2 Una colección integrada de estándares

En este artículo, [Moore, 1999] encontró que alrededor de los años, ha habido un amplio interés en crear normas para la ingeniería del software.

Un intensivo estudio descubrió aproximadamente 315 normas, guías, manuales, y otros documentos prescriptivos mantenidas por 46 organizaciones diferentes. No obstante, la captación de las normas disponibles ha estado defraudando un poco. Los usuarios esperanzados informan la dificultad encontrando las normas que satisfacen su situación particular entre los numerosos disponibles. Ellos también informan de diferencias específicas entre las normas. Por ejemplo, en una área de traslape entre dos normas, cada uno podría dar énfasis a un acercamiento diferente o podría usar una terminología diferente.

El mismo autor hace énfasis entre los beneficios tanto para el comprador, como para el vendedor, ya que contribuye a una práctica disciplinada y ello mejora la calidad del producto.

Las normas pueden ayudar a proteger al comprador debido a:

- Que provee de un vocabulario para la comunicación entre el comprador y vendedor;
- Que provee criterio objetivo que evita reclamos vagos con respecto a la naturaleza del producto.
- Que define los métodos que clasifican a las características evasivas, como la fiabilidad; y
- Que asegura al vendedor que las prácticas de aseguramiento de la calidad específicas serán aplicadas.

Cerca de los años 90's, el SESC (Software Engineering Standards Committee's) estableció un comité de planeación para iniciar el gran esfuerzo de integrar esta colección. El comité estudió las necesidades del cliente y estudió los estándares existentes, concluyendo que no había escasez de consejos disponibles para la práctica de la ingeniería del software. Sin embargo, no existía una forma clara para que los usuarios eligieran el consejo adecuado a sus necesidades. Con esta información, el autor desarrolló una arquitectura integrada para la colección SESC denominada Visión 2000. Esta colección, según [Moore, 199], comprende tres importantes criterios de organización, que se muestran en la figura no. 7.

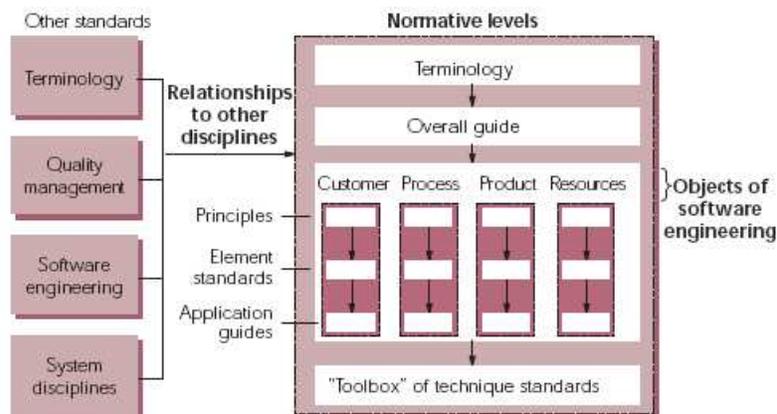


Figura no. 7. Arquitectura del Comité SES para su colección de estándares.

La arquitectura Vision 2000 es un paquete de cuatro volúmenes; cada uno, contiene estándares pertenecientes al cliente, producto, procesos ó recursos.

Estándares contenidos en el volumen del cliente

- IEEE Std. 1220, Systems Engineering Process,
- IEEE Std. 1228, Software Safety Plans,
- IEEE Std. 1233, Systems Requirements Specification, and
- IEEE Std. 1362, Concept of Operations.

Para completar el contenido, este volumen incluye el estándar de vocabulario desde la capa de terminología de la colección IEEE Std. 610.12

- Estándares contenidos en el volumen de procesos

En este volumen se encuentran los estándares que proveen detalles adicionales en muchos procesos técnicos y actividades del 12207.

- *IEEE Std. 730, Planes para el aseguramiento de la calidad del software,*
- *IEEE Std. 828, Planes para la administración de la configuración del software,*
- *IEEE Std. 1008, Pruebas unitarias del software,*
- *IEEE Std. 1012, Validación y verificación del software,*
- *IEEE Std. 1028, Revisiones del software y*
- *IEEE Std. 1219, Mantenimiento del software.*
- *IEEE/EIA 12207* trata la administración como procesos,

Así *IEEE Std. 1058, Planes para la administración de proyectos de software* está incluido a dentro del *IEEE Std. 1490, la guía PMBOK la cual provee de un amplio tratamiento de proyectos de temas de administración.*

- Estándares contenidos en el volumen del producto

Este volumen incluye los estándares utilizados para la evaluación del producto.

- IEEE Std. 1061, el cual provee una metodología para medir la calidad.
- IEEE Std. 982.1 es un diccionario de métricas que puede ser aplicado para medir la fiabilidad y características relacionadas
- IEEE Std. 1465 es una adopción del estándar ISO/IEC 12119 y provee requerimientos de calidad
- IEEE Std. 1063 se incluye en este volumen como otro estándar de documentación

- Estándares contenidos en el volumen de recursos

- IEEE Std1320.1, especificando el IDEF0,
- IEEE Std1320.2, especificando en IDEF1X97 (IDEFObject).
- IEEE Std. 1420.1 y sus suplementos provee un modelo de datos para describir e intercambiar componentes de software reusable.
- IEEE Std. 1430. Contiene estándares para herramientas CASE
- IEEE Std. 1462 considera herramientas de evaluación y selección y
- IEEE Std. 1348 considera adopción organizacional

El volumen de recursos mantiene estándares apropiados para su propósito

- IEEE Std. 830, Especificación de requerimientos de software
- IEEE Std. 829, Documentación de pruebas de software y
- IEEE Std. 1016, Descripción del diseño del software
- IEEE Std.1044 describe la clasificación de anomalías del software para una variedad de propósitos

Por último, el autor concluye diciendo que el nivel de integración entre los estándares de ingeniería del software de la IEEE no es perfecto aún. Debido a que

cada estándar está revisado de forma individual, en un lapso de alrededor de cinco años, será modificado para atacar más suavemente con el estándar IEEE/EIA 12207 y con las otras colecciones de estándares. Por supuesto, la colección IEEE continúa creciendo tanto como temas adicionales sean tratados.

CAPÍTULO III PLANTEAMIENTO DEL PROBLEMA

3.1 Descripción del Problema

Hoy en día, existe una gran incertidumbre dentro de las MiPyMes sobre la importancia de los estándares de calidad. Muchos lo relacionan con la internacionalización, pero también como pérdida de tiempo al documentar sus procesos y pérdida de dinero al certificarse. Además añaden que la mayoría de estos estándares están hechos originalmente para empresas grandes con un número considerable de empleados, a quienes se asignan los procesos establecidos en los estándares.

Sin embargo, la realidad es que las MiPyMes de software en la región presentan los siguientes problemas:

- Falta de herramientas que apoyen al cálculo de costos y tiempo de desarrollo, basándose solo en la experiencia lo que hace a la empresa dependiente de algunos empleados.
- Falta de herramientas que permitan a la MiPyMe evaluar sus procesos para determinar aquellos que sean efectivos, y encontrar puntos críticos.
- Problemas para llevar a cabo la actualización del personal.
- Problemas para aplicar tecnologías de vanguardia, lo que le resta competitividad frente a las grandes empresas de software.

- Necesidad de realizar contratos que aseguren tanto al cliente como a la empresa entrega de un producto satisfactorio y el pago del mismo en los tiempos establecidos.
- Falta de documentos oficiales que garanticen al cliente la calidad de la MiPyMe.
- Los requisitos cada vez mayores por parte de los clientes de contar con estándares de calidad para contratar desarrollos.

Estos problemas aunados a la creciente competitividad nacional, al tratado de libre comercio, y al Internet; en donde tendremos competidores que ya vienen educados en los estándares de calidad; predicen una problemática en donde solo aquellas MiPyMes que estén mejor preparadas, serán las que permanecerán en el mercado.

La necesidad de tomar en cuenta a los estándares de calidad va más allá de ganar una certificación: Es la oportunidad de adquirir metodologías de organización y planeación que aseguren la búsqueda de grandes mercados nacionales e internacionales, que contribuyan a mejorar no solo de la empresa, si no a la economía del país.

CAPÍTULO IV PLANTEAMIENTO DE LA HIPÓTESIS

4.1 Hipótesis General

La aplicación del conjunto de estándares de calidad de software del IEEE, es una opción para resolver las fallas en alguna o todas las etapas del desarrollo de sistemas de las micro y pequeña empresas de desarrollo de software, además de las áreas de desarrollo de otras empresas en la región logrando proveer con ello a las MiPyMEs de software el aseguramiento de su estadía en mercados nacionales e internacionales.

CAPÍTULO V OBJETIVOS

5.1 Objetivo General

Analizar del conjunto de los 40 estándares de calidad del software de la IEEE para seleccionar jerárquicamente los que mejor se apliquen en la micro y mediana empresa de desarrollo de software de la región.

5.2 Objetivos Específicos

1. Determinar el nivel de aplicación de los estándares de calidad en la micro y mediana empresa de desarrollo de software en la región.
2. Determinar la importancia de la calidad aplicada en el software
3. Investigar los 40 estándares de calidad de software propuestos por la IEEE
4. Seleccionar los estándares de calidad de software la IEEE propuestos a las miPymes
5. Establecer recomendaciones que permitan a la micro y mediana empresa de software y a las áreas de desarrollo de software de otras empresas, aplicar los estándares de calidad de la IEEE.

5.3 Metas

- a) Análisis de los estándares de calidad de software de la IEEE.
- b) Determinación de un modelo de aplicación de los estándares seleccionados de acuerdo al Ciclo de Vida de los Sistemas y otro modelo de los estándares seleccionados aplicados cronológicamente, por proyecto.
- c) Propuesta de una guía de aplicación de ambos modelos.

PARTE II PROPUESTA DE METODOLOGÍA

CAPÍTULO VI PROPUESTA DE SOLUCIÓN

El conjunto de estándares de software de la IEEE, proveen a las MiPyMes de software las herramientas para llevar a cabo procesos de calidad; lo que además amplía su horizonte hacia el mercado mundial. Sin embargo, no todos los estándares de ese conjunto pueden ser aplicados a todas las MiPyMes debido a los requerimientos de los mismos; lo que puede ocasionar gastos de tiempo, de recursos económicos y de personal, el descubrir cuáles son los estándares adecuados para ellos.

Es debido a esa razón que se realizó esta investigación, en donde se propondrán el uso de los estándares que sean los adecuados de acuerdo a las características de las diferentes MiPyMes de software en la región.

6.1 Análisis de los 40 estándares de software de la IEEE

El conjunto de estándares de software que presenta la IEEE se compone de 40 estándares, de los cuales la IEEE maneja al menos un documento por cada estándar, ya que existen estándares como el std. IEEE 1016-1998 que contiene 2 documentos. En total, la IEEE conforma 45 documentos en donde el documento menor contiene 13 páginas y el mayor contiene 316 páginas (ver figura no. 8), lo que resulta una cantidad de 2, 647 páginas que las MiPyMes tendrían que leer para determinar cuál estándar es apto para su empresa.

Es ahí en donde radica la importancia de esta investigación, ya que el desarrollador al leer el presente trabajo, encontrará que estándares deberá leer dependiendo del tipo de empresa que tenga, lo que le ahorrará tiempo, dinero y recursos humanos.

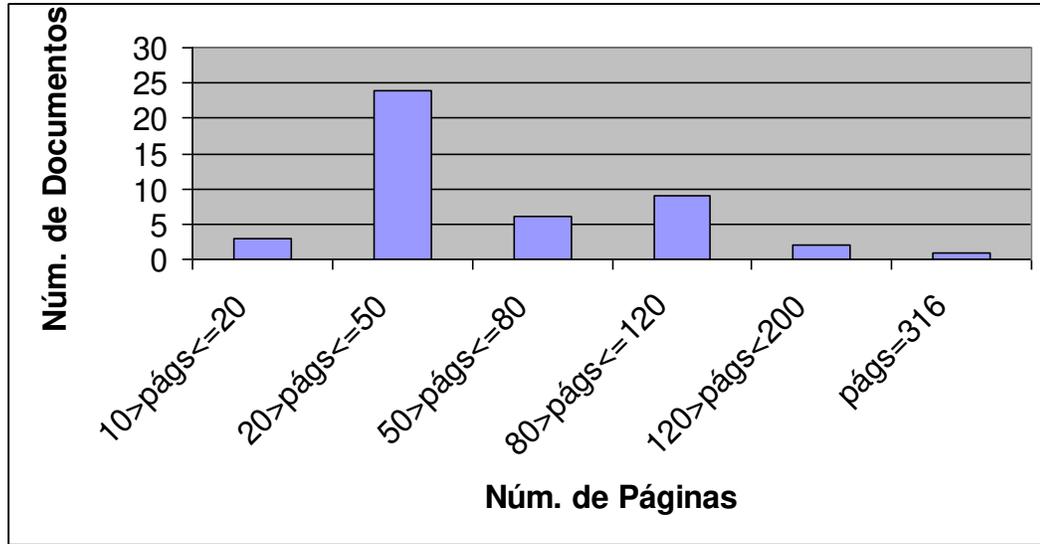


Figura 8. Relación de páginas por documentos de los estándares de software de la IEEE.

6.1.1 Organización de la colección de estándares de la ingeniería de software de la IEEE

Existen diferentes colecciones del conjunto de estándares de software de la IEEE dentro de la Guía SWEBOK (Software Engineering Body of Knowledge), la cual ha organizado el conjunto de estándares por áreas de conocimiento, sin embargo; provee las siguientes organizaciones alternativas:

- Organización por Área de Conocimiento.
- Organización por Cuatro Objetos.
- Organización por Procesos
- Organización por Temas

6.1.1.1 Organización por Área de Conocimiento.

Cada tema resume uno de las diez áreas de conocimiento del SWEBOK (así como dos áreas adicionales):

- Requerimientos de Software
- Diseño de Software

- Construcción de Software
- Pruebas de Software
- Mantenimiento de Software
- Administración de la Configuración del Software
- Administración de la Ingeniería del Software
- Procesos de Ingeniería de Software
- Métodos y Herramientas de Ingeniería de Software
- Calidad de Software
- Ingeniería de Sistemas de Software
- Mejores Prácticas en Internet

6.1.1.2 Organización por Cuatro Objetos

La edición colectada de 1999 del conjunto de estándares estuvo organizada en una manera diferente. Esa organización aplicó dos principios al grupo de estándares: Niveles de prescripción y objetos de la ingeniería de software.

6.1.1.3 Organización por Procesos

Otra forma de organizar la colección es usando un modelo de referencia de procesos. El conjunto de estándares de software ha proveído de un modelo de referencia de procesos basado en los procesos del IEEE/EIA 12207.0 y algunos estándares complementarios.

6.1.1.4 Organización por temas.

La edición de 1997, explicó que la colección fue organizada por temas. Ocho temas fueron usados:

- Documentación.
- Procesos del Ciclo de vida
- Mediciones (Incluyendo fiabilidad)
- Planes
- Administración de Proyectos
- Terminologías
- Herramientas

6.2 Perfil de las empresas de Software en México

México cuenta con una posición favorable para convertirse en un competidor de talla mundial en este ramo del software, gracias a su ubicación geográfica, perfil demográfico y estado de desarrollo tecnológico. No obstante el potencial de desarrollo es evidente, la industria del software es apenas incipiente en nuestro país: participa con tan sólo el 0.10% del PIB (cifras de 2000). Aunque no existe un padrón exhaustivo de esta industria que proporcione información exacta, una muestra de 206 empresas desarrolladoras de software muestra el perfil actual de la industria que es mayoritariamente micro y pequeña, con un tamaño muy inferior al del promedio internacional, que es de 250 empleados.

Tabla no. 4 Empresas desarrolladoras de software en México

Tamaño	Número de empleados	Promedio de empleados	Número de empresas
Micro	Menos de 15	7	63
Pequeña	De 16 a 100	60	117
Mediana	De 101 a 250	175	14
Grande	De 251 a 1,000	600	11
Corporativa	Más de 1,000	1,500	1
Total			206

6.3 Propuesta de Aplicación.

Según [Santillán, 2005] los principales retos para las MiPyMes de software en México estriban en entrar a la competencia mundial, contar con una planeación financiera, estratégica y de marketing. El uso de estándares de calidad, ayudarán a que estos retos se conviertan en logros alcanzables.

Sin embargo, como se explicó anteriormente, no todos los estándares pueden ser aplicados a todas las MiPyMes, y estas gastarían recursos económicos y humanos para determinar cuáles son los estándares adecuados para ellos.

En el evento de prosoft celebrado los días 23 y 24 de noviembre del 2005 en la ciudad de México, los integrantes de las MiPyMes mostraron interés en hacer uso de todas las herramientas que les permitan mejorar sus procesos, sin embargo también expusieron su temor al ver los gastos que se requerían para lograrlo.

Para ayudar a estos empresarios a reducir sus costos, se les presenta en este documento; una propuesta de los estándares adecuados a las MiPyMes de la región, proponiéndoles modelos de acuerdo al tipo de MiPyme

CAPÍTULO VII ANÁLISIS Y CLASIFICACIÓN

La metodología en la que se desarrolló la presente investigación, fue enriquecida por el programa federal Prosoft, el cual está enfocado a las miPymes de desarrollo de software del país. Apoyando su crecimiento a través de financiamientos y proponiendo la utilización del primer estándar de calidad mexicano MoProsoft, para la internalización de sus productos. Este programa respalda la importancia la aplicación de estándares de calidad en los procesos que utilizan las MiPymes para el desarrollo de software.

A continuación se describe la metodología y los estándares de calidad de software de la IEEE clasificados por áreas.

7.1 Selección de la clasificación

Después de estudiar las diferentes clasificaciones que maneja la IEEE en sus estándares de calidad de software, para esta investigación en particular se decidió utilizar la clasificación por área de conocimiento, ya que esta maneja los procesos que la mayoría de las MiPyMes de la región utiliza. Por lo que esta clasificación les facilitará la elección del modelo que requieran.

7.2 Descripción de los estándares de calidad de software de la IEEE

A continuación se listan el conjunto de estándares de calidad de software que establecidos por la IEEE clasificados por el área de conocimiento.

7.2.1 Área de conocimiento de Requerimientos de Software.

Según [IEEE, 2004], un requerimiento es definido como una propiedad que debe ser exhibida para resolver algún problema del mundo real.

La primer subárea de conocimiento es el proceso de la ingeniería de requerimiento. Describe modelos de procesos, actores de procesos, soporte de procesos y administración, además de la mejora de la calidad de los procesos.

La segunda subárea es la licitación de requerimientos, la cual se preocupa por el origen de los requisitos y como pueden ser recolectados por la ingeniería de requerimientos. Incluye las fuentes del requerimiento y las técnicas para la licitación.

La tercer subárea es el análisis de requerimientos, que se preocupa por los procesos del análisis de requerimientos para:

- Detectar y resolver conflictos entre requerimientos.
- Descubrir los límites del sistema y como este debe interactuar con el medio ambiente.
- Elaborar requerimientos de sistemas para requerimientos de software.

La cuarta subárea es la especificación de requerimientos. Que describe la estructura, calidad y fiabilidad del documento de requerimientos.

La quinta subárea se refiere a la validación de requerimientos. Su objetivo es recoger cualquier problema antes de que los recursos sean comprometidos al dirigirse a los requerimientos. La validación de requerimientos se preocupa de los procesos de examinar el documento de requerimientos que asegura que el sistema está definido correctamente.

La última subárea contiene la administración de requerimientos, la cual es una actividad que abarca todo el ciclo de vida del software. Fundamentalmente trata de cambios de administración, y el mantenimiento de los requerimientos en un estado que con precisión refleja el software que es o ha sido construido.

Tabla no. 5. Listado de Estándares que corresponden al área de Requerimientos de Software

Estándar	Título	Resumen
IEEE Std 830™-1998	Prácticas Recomendadas por la IEEE para la Especificación de Requerimientos de Software	El contenido y calidad de una buena especificación de requerimientos (SRS) son descritas en diferentes marcos de ejemplos. Esta práctica recomendada también puede ser aplicada para apoyar la selección de productos de software casero y comercial. Provee además las directrices para el cumplimiento con el estándar IEEE/EIA 12207.1-1997.
IEEE Std 1062™, Edición 1998	Práctica recomendada por la IEEE para la adquisición de software	Describe un conjunto de prácticas de calidad útiles que pueden seleccionarse y pueden aplicarse durante uno o más pasos en los procesos de adquisición de software. Esta práctica puede ser aplicada al software que corra en cualquier sistema de computadora indiferente al tamaño o complejidad del software, pero está más preparado para su uso sobre modificaciones externas al software y a software completamente desarrollado.
IEEE Std 1228™-1994	Estándar de la IEEE para los planes de seguridad del software	Se establecen los requerimientos mínimos aceptables para un plan de seguridad del software. Este estándar aplica el plan para usarlo en el desarrollo, procuración, mantenimiento y retiro de software de seguridad crítica. Este estándar requiere que el plan sea preparado dentro del contexto del programa de seguridad del sistema. Solo son incluidos aspectos de seguridad del software. No contiene provisiones especiales requeridas para el software usadas en sistemas distribuidos o procesamientos en paralelo.
IEEE Std 1233™, Edición 1998	Guía de la IEEE para las Especificaciones de Requerimientos en el Desarrollo de Sistemas	Orientación para el desarrollo del conjunto de requerimientos, Especificación de Requerimientos del Sistema (SyRS), que procurará que una necesidad expresada sea proporcionada. El desarrollar una SyRS

		<p>incluye la identificación, organización, presentación y modificación de requerimientos. También están dirigidas las condiciones para incorporar conceptos operacionales, diseño de restricciones y diseño de requerimientos de configuración dentro de la especificación. Esta guía también cubre las características necesarias y cualidades de requerimientos individuales y el conjunto de todos los requerimientos.</p>
IEEE Std 1320.1™-1998	Estándar IEEE para la sintaxis del lenguaje de modelado funcional y semánticas para IDEF0	<p>La función de modelado IDEF0 está diseñada para representar las decisiones, acciones y actividades de una organización existente o prospecto, ó sistema. Las gráficas y textos de acompañamiento son presentados en una forma organizada y sistemática para ganar entendimiento, análisis de apoyo, lógica para cambios potenciales, requerimientos específicos, soporte a nivel sistema de diseño y de integración de actividades. IDEF0 puede ser usada para modelar una amplia variedad de sistemas, compuestos de personas, máquinas, materiales, computadoras, y de información de todas las variedades y estructuradas por las relaciones entre ellas; tanto con las automatizadas, como con las que no.</p>
IEEE Std 1320.2™-1998	Estándar IEEE para la sintaxis y la semántica del Lenguaje de Modelado Conceptual para IDEF1X ₉₇ (IDEF _{object})	<p>IDEF1X₉₇ consiste de dos lenguajes de modelado conceptual. El estilo principal que soporta el lenguaje de modelado de información/datos y es compatible de forma descendente con el estándar 1993 del gobierno de EU, FIPS PUB 184. El lenguaje de Identidad de estilos está basado sobre el modelo objeto con reglas declarativas y restricciones. IDEF1X₉₇ identifica estilos e incluye las estructuras para los distintos pero relacionados componentes objeto de abstracción: Interfaces, peticiones, y realización; utiliza gráficos para establecer la interfase; y define una declarativa, Lenguaje de restricciones y reglas directamente ejecutables para la realización de peticiones.</p>
IEEE Std 1362™-1998	Guía IEEE para Definición de conceptos de Sistemas de Tecnologías de Información, de documentos de operación	<p>Se describen los documentos de formato y contenidos de conceptos de operaciones (ConOps). Un ConOps es un documento orientado al usuario que describe características de sistemas para un sistema propuesto desde el punto de vista del usuario. El documento</p>

	(ConOps)	es usado para comunicar el conjunto de características del sistema cuantitativamente y cualitativamente al usuario, comprador, desarrollador y otros elementos organizacionales. Es usado para describir al usuario la(s) organización(es), misión(es), y objetivos organizacionales desde un punto de vista de sistemas integrados.
IEEE Std 1465™-1998	Estándar IEEE. Adopción del estándar internacional ISO/IEC 12119:1994(E) Tecnologías de Información para requerimientos y pruebas de Calidad en Paquetes de Software.	Se establecen los requerimientos de calidad para paquetes de software e instrucciones de cómo probar un software empaquetado contra esos requerimientos. Los requerimientos aplicados para el software empaquetado son ofrecidos y entregados, no los procesos de producción (incluyendo actividades y productos intermedios como las especificaciones).
IEEE Std 14143.1™-2000	Adopción IEEE de ISO/IEC 14143-1:1998 Tecnologías de Información para la Medición del software, Medición del tamaño funcional. Parte 1: Definición de Conceptos.	Esta parte (volumen) del ISO/IEC 14143 define los conceptos fundamentales para la Medición del Tamaño Funcional (FSM) y describe los principios generales para aplicar el método FSM. Esta parte del ISO/IEC 14143 no provee las reglas detalladas de cómo: Medir el Tamaño Funcional de software usado un método particular; Usa los resultados obtenidos desde un método particular; seleccionar un método particular. Esta parte de ISO/IEC 14143 es aplicable cuando se determina si el método para establecer el tamaño del software es un método FSM. Este no previene al desarrollo de varios métodos, más bien provee de una base para evaluar si un particular método conforma el FSM. La implementación provee la relación de la interpretación IEEE del ISO/IEC 14143-1:1998.

7.2.2 Área de conocimiento de Diseño de Software.

De acuerdo a la IEEE, el diseño de software es una actividad que abarca en su totalidad el ciclo de vida del software. Trata fundamentalmente de cambios de administración y el mantenimiento de los requerimientos en un estado acertadamente refleja al software para ser o que ha sido construido. El área de conocimiento está dividida dentro de seis subáreas (ver tabla no. 6).

La primera presenta los conceptos básicos y nociones que forman una base subrayada para el entendimiento de la regla y el alcance del diseño de software. Estos son generalmente los conceptos, el contexto del diseño de software, los procesos de diseño, y las técnicas que permiten el diseño del software.

La segunda subárea reagrupa los principales problemas del diseño del software. Estas incluyen la entrada, control y manejo de eventos, distribución, error y manejo de excepciones, sistemas interactivos y persistencia.

La tercer subárea es la estructura y arquitectura, en particular, la estructura arquitectónica y puntos de vista, estilos arquitectónicos, diseño de patrones, y familias de programas y marcos.

La cuarta subárea describe el análisis de calidad de diseño del software y la evaluación. Mientras el total de área del conocimiento es leal a la calidad del software, esta subárea presenta los temas más específicamente relacionados al diseño del software. Estos aspectos son atributos de calidad, análisis de calidad y herramientas de evaluación y medición.

Tabla no. 6. Listado de Estándares que corresponden al área de Diseño de Software

Estándar	Título	Resumen
IEEE Std 982.1™-1988	Estándar IEEE Diccionario de Medidas para producir software fiable	Este estándar provee de un conjunto de medidas indicativas de la fiabilidad del software que puede ser aplicado tanto a productos de software como al desarrollo y procesos de soporte. Fue motivado por la necesidad de los desarrolladores de software y usuarios quienes están confrontados con una plétora de modelos, técnicas y medidas. Hay una necesidad para medidas que pueden ser

		<p>aplicadas cercanamente en el desarrollo de procesos que pueden ser indicadores de la fiabilidad del producto entregado. Este estándar provee una común y consistente definición de un conjunto de medidas que pueden conocer aquellas necesidades.</p>
IEEE Std 1016™-1998	IEEE Prácticas recomendadas para las descripciones de diseño de software	<p>La información necesaria contenida y recomendaciones para una organización para el diseño de descripciones de software (SDDs) está descrita. Un SDD es una representación de un sistema de software que es usada como un medio para comunicar información de diseño de software. Esta práctica recomendada es aplicable a documentos en papel, bases de datos automatizadas, lenguajes de descripción de diseños, u otros significados de descripción.</p>
IEEE Std 1028™-1997	Estándar IEEE para revisiones de software	<p>Este estándar define cinco tipos de revisiones de software, en conjunto con los procedimientos requeridos para la ejecución de cada tipo de revisión. Este estándar es concerniente solo con estas revisiones; no define procedimientos para determinar la necesidad de una revisión, no especifica la disposición de los resultados de la revisión. Los tipos de revisión incluyen revisiones de administración, revisiones técnicas, inspecciones y auditorías.</p>
IEEE Std 1320.1™-1998	Estándar IEEE para la sintaxis y semántica del Lenguaje de Modelado Funcional para IDEF0	<p>Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de Software.</p>
IEEE Std 1320.2™-1998	Estándar IEEE para la sintaxis y la semántica del Lenguaje de Modelado Conceptual para IDEF1X ₉₇ (IDEF _{object})	<p>Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de Software.</p>
IEEE Std 1471™-2000	IEEE Prácticas recomendadas para la descripción arquitectónica de sistemas intensivos de software	<p>Esta práctica recomendada dirige las actividades de la creación, análisis y sostenimiento de las descripciones arquitectónicas. Se establece un marco conceptual para esta descripción. Se describe el contenido de una descripción arquitectónica. Los anexos proveen la razón para los conceptos llave y terminologías, la relación con los otros estándares y ejemplos de uso.</p>

7.2.3 Área de conocimiento de Construcción de Software.

La construcción del software es un acto fundamental de la ingeniería de software: La construcción de software significativo a través de una combinación de código, validación y pruebas (pruebas de unidad).

El primer método y más importante para dividir el tema de la construcción del software dentro de pequeñas unidades es reconocer los cuatro principios que afectan más fuertemente la forma en la que el software es construido. Estos principios son la reducción de la complejidad, la anticipación a la adversidad, la estructuración para la validación, y el uso de estándares externos.

Un segundo y menos importante método de dividir el tema de Construcción de software en pequeñas unidades es el de organizar tres métodos-estilos de construcción de software, llamada: Lingüística, Formal y Visual.

Tabla no. 7. Listado de Estándares que corresponden al área de Construcción de Software

Estándar	Título	Resumen
IEEE Std 829™-1998	Estándar IEEE para la documentación de prueba del software	Un conjunto de documentos de prueba de software básico es descrito. Este estándar especifica la forma y contenido de documentos de prueba individual. No especifica el conjunto requerido de estos documentos.
IEEE Std 982.1™-1988	Estándar IEEE Diccionario de Medidas para producir software fiable	Nota: Se encuentra descrito en el área de conocimiento de Diseño de Software.
IEEE Std 1008™-1987(R1993)	Estándar IEEE para Pruebas de unidad de Software	El principal objetivo de este estándar es especificar un estándar aproximado a la unidad de software que puede ser usado como base para práctica de ingeniería de software legítimo. Un segundo objetivo es describir los conceptos de ingeniería de software y probar suposiciones en las cuales son basadas estas aproximaciones. Un tercer objetivo es el de proporcionar una guía e información para ayudar junto con la implementación y uso del estándar de aproximación de pruebas de unidad.
IEEE Std 1028™-1997	Estándar IEEE para Revisiones de Software	Nota: Se encuentra descrito en el área de conocimiento de Diseño de Software.
IEEE Std 1061™-1998	Estándar IEEE para Metodologías de Métricas de Calidad de Software	Una metodología para establecer requerimientos de calidad e identificar, implementar, analizar y validar los procesos y son definidas métricas de calidad de software. La metodología abarca la totalidad del ciclo de vida.
IEEE Std 1063™-2001	Estándar IEEE para la Documentación de usuarios de Software	El mínimo de requerimientos para la estructura, el contenido de la información, y el formato de la documentación del usuario, incluyendo tanto documentos impresos como electrónicos usados en el ambiente de trabajo por los usuarios de sistemas, son proveídos en este estándar.
IEEE Std 1517™-1999	Estándar IEEE para Procesos de Reuso en los procesos del Ciclo de Vida para Tecnologías de Información.	Un marco común para extender los procesos del ciclo de vida del estándar IEEE/EIA 12207.0-1966 para incluir la práctica sistemática que proporciona el reuso del software. Este estándar especifica los procesos, actividades, tareas para ser aplicadas durante cada etapa del ciclo de vida para permitir que

un producto de software sea construido desde valores reusables. También especifica los procesos, actividades, y tareas que permitan la identificación, construcción, mantenimiento y administración de valores suministrados.

7.2.4 Área de conocimiento de Prueba de Software.

La prueba de software consiste en la verificación dinámica más allá de un programa o un conjunto finito de casos, seleccionados adecuadamente desde el dominio de ejecuciones normalmente infinitas, en contra de lo especificado. Incluye cinco subáreas.

Comienza con una descripción de conceptos básicos. Primero se presenta la terminología de pruebas, después son descritas las fundamentaciones teóricas de pruebas, junto con la relación de pruebas de otras actividades.

La segunda subárea son los niveles de prueba. Están divididos entre las metas y los objetivos de prueba.

La tercer subárea es las técnicas de autopuebas. Una primera categoría es agrupada sobre criterios de la base sobre la cual las pruebas son generadas, y un segundo grupo basado sobre la ignorancia del conocimiento de implementación. Se presenta una discusión de cómo seleccionar y combinar las técnicas apropiadas.

La cuarta subárea cubre mediciones de prueba relacionadas. Las mediciones están agrupadas dentro de aquellas relacionadas a la evaluación del programa bajo prueba y la evaluación de las pruebas desempeñadas.

La última subárea describe la administración específica a los procesos de prueba. Incluye administración importante y actividades de prueba.

Tabla no. 8. Listado de Estándares que corresponden al área de Pruebas del Software

Estándar	Título			Resumen
IEEE Std 829™-1998	Estándar de documentación de software	IEEE para la prueba de	la	Nota: Se encuentra descrito en el área de conocimiento de Construcción de Software.
IEEE Std 982.1™-1988	Estándar de medidas para producir software reutilizable.	IEEE Diccionario de	software	Nota: Se encuentra descrito en el área de conocimiento de Diseño de Software.
IEEE Std 1008™-1987 (R1993)	Estándar de unidad de Software	IEEE para Pruebas de	Software	Nota: Se encuentra descrito en el área de conocimiento de Construcción de Software.
IEEE Std 1044™-1993	Estándar de Anomalías de software	IEEE Clasificación para	Anomalías de software	Un informe de acercamiento a la clasificación de anomalías encontradas en el software y se proporciona su documentación. Se describe el proceso de anomalías descubierta durante cualquier fase del ciclo de vida, se proporciona una lista clasificada de anomalías y elementos relacionados que son útiles para identificar y rastrear las anomalías. Este estándar no intenta definir procedimientos o formatos de requerimientos para usar el esquema de clasificación. Identifica algunas medidas de clasificación y no intenta definir todos los datos que soportan el análisis de una anomalía.
IEEE Std 1063™-2001	Estándar de Documentación de Software	IEEE para la Documentación de usuarios de	Software	Nota: Se encuentra descrito en el área de conocimiento de Construcción de Software.
IEEE Std 1228™-1994	Estándar de planes de seguridad del software	de la IEEE para los	planes de seguridad del software	Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de Software.

7.2.5 Área de conocimiento de Mantenimiento de Software.

Una vez en operación, las anomalías son descubiertas, operando cambios al ambiente, y aparecen nuevos requerimientos de usuarios. La fase de mantenimiento del ciclo de vida comienza después de la entrega, pero las actividades de mantenimiento ocurren mucho antes. El área de conocimiento de software está dividida en seis subáreas.

La primera presenta los conceptos básicos de dominio, definiciones y las actividades principales y problemas de mantenimiento de software.

La segunda subárea describe los procesos de mantenimiento, basados en el estándar IEEE 1219 y ISO/IEC 14764.

La tercer subárea reagrupa los principales problemas relacionados con el mantenimiento de software. Los tópicos cubiertos son técnica, administración, costo y estimación y problemas de medición.

Las técnicas para mantenimiento constituyen la cuarta subárea. Estas técnicas incluyen comprensión de programas, reingeniería, ingeniería inversa y análisis de impacto.

Tabla no. 9. Listado de Estándares que corresponden al área de Mantenimiento de Software

Estándar	Título		Resumen
IEEE Std 1219™-1998	Estándar de mantenimiento de software	IEEE para	Se describen los procesos de mantenimiento y actividades de mantenimiento para la ejecución de software.
IEEE Std 1061™-1998	Estándar Metodologías de	IEEE Métricas de	para Nota: Se encuentra descrito en el área de conocimiento de Construcción de Software.

IEEE Std 14143.1™-2000 Adopción IEEE de ISO/IEC 14143-1:1998 Tecnologías de Información para la Medición del software, Medición del tamaño funcional. Parte 1: Definición de Conceptos. Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de Software.

7.2.6 Área de conocimiento de Administración de la Configuración de Software.

La administración de la configuración del software (SCM) es la disciplina de identificar la configuración de un sistema en distintos puntos de tiempo para los propósitos de cambios de control sistemáticos a la configuración del software, manteniendo la integridad y trazabilidad de la configuración a través del ciclo de vida del sistema. Esta área de conocimiento incluye seis subáreas.

La primer subárea son los procesos de administración SCM. Cubre los temas del contexto organizacional para SCM, las restricciones y dirección para SCM, planeación para SCM, el plan mismo de SCM, y vigilancia de SCM.

La segunda subárea es la identificación de la configuración del software, la cual identifica los elementos a ser controlados, establece esquemas de identificación para los elementos y sus versiones, y establece las herramientas y técnicas a ser usadas en la adquisición y administración de elementos controlados. Los temas en esta subárea son la identificación de los elementos a ser controlados y la biblioteca de software.

La tercer subárea es el control de configuración del software, la cual es la administración de cambios durante el ciclo de vida del software. Los temas son:

- Solicitud, evaluación y aprobación de cambios de software,
- Implementación de cambios de software y
- Concesión de desviaciones y renuncia

La cuarta subárea es la contabilidad del estado de la configuración del software. Estos temas están en la información del estado de la configuración del software y el reporte de estado.

La quinta subárea es la auditoria de la configuración del software. Consiste en la auditoria de la configuración física del software, y auditorias internas a procesos de una línea de fondo de software.

La última subárea es la administración de la liberación del software y entrega, cubriendo la construcción del software y la administración de la liberación del software.

Tabla no. 10. Estándares correspondientes al área de Admón. de la Configuración del Software

Estándar	Título	Resumen
IEEE Std 730™-2002	Estándar IEEE para Planes de aseguramiento de la calidad del software	El estándar especifica el formato y contenido en el plan de aseguramiento de la calidad del software. Se encuentra en el IEEE/EIA 12207.1 Requerimientos que así se planean.
IEEE Std 828™-1998	Estándar IEEE para los planes de administración de configuración del software.	Son establecidos los requerimientos mínimos requeridos contenidos en el Plan de administración de configuración del software (SCMP), y se definen las actividades a ser dirigidas y sus requerimientos para cualquier porción del ciclo de vida del software.
IEEE Std 1028™-1997	Estándar IEEE para revisiones de software.	Nota: Se encuentra descrito en el área de conocimiento de Diseño de Software.

7.2.7 Área de conocimiento de Administración de la Ingeniería de Software.

Mientras sea verdad decir que en un sentido debería ser posible administrar la ingeniería del software en la misma forma que cualquier otro proceso (complejo), existen aspectos particulares de productos de software y procesos de ingeniería de software que complican la administración efectiva. Hay tres subáreas para la administración de la ingeniería de software.

La primera es la administración organizacional, administración de políticas de comprensión, administración de personal, administración de la comunicación, administración de la cartera y administración de la adquisición.

La segunda subárea es la administración de procesos de proyectos, incluyendo la iniciación y la definición del alcance, difusión, revisión y evaluación y cierre.

La última subárea es medición de ingeniería de software, donde los principios generales acerca de las mediciones de software son cubiertos. Los primeros temas presentados son los objetivos y un programa de medición, seguido de la selección de la medición, software de medición y su desarrollo, colección de datos y modelos de métricas de software.

Tabla no. 11. Estándares correspondientes al área de Administración de la Ingeniería de Software

Estándar	Título	Resumen
IEEE Std 730™-2002	Estándar IEEE para Planes de aseguramiento de la calidad del software	Nota: Se encuentra descrito en el área de conocimiento de Administración de la Configuración de Software.
IEEE Std 828™-1998	Estándar IEEE para los planes de administración de configuración del software.	Nota: Se encuentra descrito en el área de conocimiento de Administración de la Configuración de Software.

IEEE Std 982.1™-1988	Estándar IEEE Diccionario de medidas para producir software reutilizable.	Nota: Se encuentra descrito en el área de conocimiento de Diseño de Software.
IEEE Std 1028™-1997	Estándar IEEE para revisiones de software.	Nota: Se encuentra descrito en el área de conocimiento de Diseño de Software.
IEEE Std 1044™-1993	Estándar IEEE Clasificación para Anomalías de software	Nota: Se encuentra descrito en el área de conocimiento de Pruebas de Software.
IEEE Std 1045™-1992	Estándar IEEE para métricas de productividad de Software	Se define una forma consistente para medir los elementos que van dentro de la productividad del software. La terminología de métricas de productividad de software es dada para asegurar un entendimiento de medición de datos tanto del código fuente como de la producción de documentos. Si bien este estándar preescribe medidas para caracterizar los procesos de software, No establece normas de productividad del software, no recomienda medidas de productividad como un método para evaluar proyectos de desarrollo de software. Este estándar no mide la calidad del software. No reclama mejorar la productividad, solo la mide. El objetivo de este estándar es mejorar el entendimiento de los procesos de software, el cual puede dar una idea para mejorarlos.
IEEE Std 1058™-1998	Estándar IEEE para Planes de administración de proyectos de software.	Se describen el formato y contenido del plan de administración de software, aplicables a cualquier tipo o tamaño del proyecto. Son identificados los elementos que podrían aparecer en todo el plan.
IEEE Std 1061™-1998	Estándar IEEE para Metodologías de Métricas de Calidad de Software	Nota: Se encuentra descrito en el área de conocimiento de Construcción de Software.
IEEE Std 1062™, 1998 Edition	Práctica recomendada por la IEEE para la adquisición de software	Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de Software.
IEEE Std 1219™-1998	Estándar IEEE para el mantenimiento del software	Nota: Se encuentra descrito en el área de conocimiento de Mantenimiento de Software.
IEEE Std 1228™-1994	Estándar de la IEEE para los	Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de

			planes de seguridad del software	Software.
IEEE Std 1490™-1998	Guía estándar de administración de proyectos de Conocimiento	IEEE Adopción del PMI Una Guía al cuerpo de administración de proyectos		El subconjunto del cuerpo de conocimiento de administración de proyectos generalmente aceptada es identificado y descrito en esta guía. "Generalmente aceptada" significa que el conocimiento y prácticas descritas son aplicables a la mayoría de proyectos la mayoría del tiempo, y que es el acuerdo general extendido acerca de su valor y utilidad. No significa que el conocimiento y prácticas deberían ser aplicados uniformemente a todos los proyectos sin considerar si ellos son apropiados.
IEEE Std 1540™-2001	Estándar Administración de procesos del ciclo de vida del software	IEEE para Administración de Riesgos en los procesos del ciclo de vida del software		Se define un proceso para la administración de riesgos en el ciclo de vida del software. Puede ser añadido al conjunto existente de procesos del ciclo de vida definidos por las series de estándares IEEE/EIA 12207, o puede ser usada independientemente.
IEEE Std 14143.1™-2000	Adopción de Información para la Medición del software, Medición del tamaño funcional. Parte 1:	IEEE de ISO/IEC 14143-1:1998 Tecnologías de		Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de Software.

7.2.8 Área de conocimiento de Procesos de Ingeniería de Software.

El área de conocimientos de procesos de ingeniería de software se preocupa por la definición, implementación, medición, cambio y mejora de los mismos procesos de la ingeniería de software. Está dividido en seis subáreas.

La primera presenta los conceptos básicos: Temas y terminologías.

La segunda subárea presenta la infraestructura de procesos, donde se describen conceptos agrupados de los procesos de ingeniería de software, como fábrica de experiencias.

La tercer subárea trata con mediciones específicas para procesos de ingeniería de software. Presenta la metodología y paradigmas de medición en el campo.

La cuarta subárea describe el conocimiento relacionado a la definición de procesos: Los varios tipos de definiciones de procesos, modelos estructura de ciclos de vida, modelos del ciclo de vida del software, las notaciones usadas para representar estas definiciones, métodos de definiciones de procesos y automatización relativa a las diferentes definiciones.

La quinta subárea presenta análisis de procesos cualitativos, especialmente la revisión de definición de procesos y análisis de causa raíz.

Finalmente, la sexta subárea concluye con la implementación de procesos y cambios. Describe los paradigmas y guías para la implementación de procesos y cambios y la evaluación del resultado de la implementación y cambio.

Tabla no. 12. Listado de Estándares que corresponden al área de Procesos de Ingeniería de Software

Estándar	Título	Resumen
IEEE Std 1008™-1987 (R1993)	Estándar IEEE para Pruebas de unidad de Software	Nota: Se encuentra descrito en el área de conocimiento de Construcción de Software.
IEEE Std 1044™-1993	Estándar IEEE Clasificación para Anomalías de software	Nota: Se encuentra descrito en el área de conocimiento de Prueba de Software.
IEEE Std 1045™-1992	Estándar IEEE para métricas de productividad de Software	Nota: Se encuentra descrito en el área de conocimiento de la Administración de la Ingeniería de Software.
IEEE Std 1061™-1998	Estándar IEEE para Metodologías de Métricas de Calidad de Software	Nota: Se encuentra descrito en el área de conocimiento de Construcción de Software.
IEEE Std 1074™-	Estándar IEEE para el Desarrollo	Nota: Se encuentra descrito en el área

1997	de procesos del ciclo de vida del software.	de conocimiento de Administración de la Ingeniería de Software de Software.
IEEE Std 1219™-1998	Estándar IEEE para el mantenimiento del software	Nota: Se encuentra descrito en el área de conocimiento de Mantenimiento de Software.
IEEE Std 1517™-1999	Estándar IEEE para Tecnologías de Información Procesos de Reuso en Procesos del Ciclo de Vida del Software	Nota: Se encuentra descrito en el área de conocimiento de Construcción de Software.
IEEE Std 1540™-2001	Estándar IEEE para Administración de Riesgos en los procesos del ciclo de vida del software	Nota: Se encuentra descrito en el área de conocimiento de Administración de la Ingeniería de Software.
IEEE/EIA 12207.0-1996	Implementación Industrial del estándar internacional ISO/IEC 12207:1995. Estándar para Tecnologías de Información Procesos del Ciclo de Vida del Software	ISO/IEC 12207 provee un marco común para desarrollar y administrar el software. IEEE/EIA 12207.0 consiste de las aclaraciones, adiciones y cambios aceptados por la IEEE y la EIA como las formuladas por un proyecto unido de las dos organizaciones. IEEE/EIA 12207.0 contiene conceptos y guías para adoptar mejor el entendimiento y aplicación del estándar. Así este estándar provee a la industria una base para prácticas del software que serían usadas para negocios tanto nacionales como internacionales.
IEEE/EIA 12207.1™-1996	Implementación Industrial del Estándar ISO/IEC 12207:1995. Estándar para Tecnologías de Información –Procesos del Ciclo de vida del Software- Datos del ciclo de vida.	ISO/IEC 12207 provee un marco común para desarrollar y administrar software. IEEE/EIA 12207.0 consiste de las aclaraciones, adiciones, y cambios aceptados por la IEEE y la EIA como las formuladas por un proyecto unido de las dos organizaciones en conjunto. IEEE/EIA 12207.1 proporciona guías para grabar los datos resultantes del ciclo de vida desde los procesos de ciclo de vida de IEEE/EIA 12207.0
IEEE/EIA 12207.2-1997	Implementación Industrial del Estándar Internacional ISO/IEC 12207:1995 Estándar de Tecnologías de Información – Procesos del Ciclo de vida- Consideraciones de Implementación.	ISO/IEC 12207 provee un marco común para desarrollar y administrar software. IEEE/EIA 12207.0 consiste de las aclaraciones, adiciones, y cambios aceptados por la IEEE y la EIA como las formuladas por un proyecto unido de las dos organizaciones en conjunto. IEEE/EIA 12207.2 proporciona una guía de consideración en la implementación para las cláusulas normativas del IEEE/EIA 12207.0. La guía está basada sobre la experiencia de la industria de

7.2.9 Área de conocimiento de Herramientas de Ingeniería de Software y Métodos.

Las herramientas de Ingeniería de Software y el área de conocimiento de métodos están incluidas tanto en los ambientes de desarrollo de software como en los métodos de desarrollo.

Los ambientes de desarrollo de software son herramientas basadas en computadoras que son planeadas para asistir a los procesos de desarrollo de software. Los métodos de desarrollo imponen la estructura sobre las actividades de desarrollo de software con la meta de hacer la actividad sistemática y al final, más probablemente para tener éxito.

La sección particionada de herramientas de software usa la misma estructura que el total de la guía para el cuerpo de conocimiento de ingeniería de software. Las primeras nueve subsecciones corresponden a las otras nueve áreas de conocimiento. Se adicionan otras dos subsecciones: Una para herramientas de soporte a infraestructura que no le queda a ninguna sección cercana, y una subsección miscelánea para temas, como técnicas de integración de herramientas, que son potencialmente aplicables para toda clase de herramientas.

La sección de métodos de desarrollo de software está dividida dentro de cuatro subsecciones: métodos heurísticos tratando con aproximaciones informales, métodos formales tratando con aproximaciones basadas matemáticamente, métodos de prototipo tratando con aproximaciones de desarrollo de software basados en varias formas prototipo y miscelánea de problemas de métodos.

Tabla no. 13. Estándares correspondientes al área de Herram. de Ingeniería de Software y Métodos.

Standard:	Título	Resumen
IEEE Std 1028™-1997	Estándar IEEE para Revisones de Software	Nota: Se encuentra descrito en el área de conocimiento de Diseño de Software.
IEEE Std 1175.1™-2002	Guía IEEE para Interconexión de Herramientas CASE. Clasificación y Descripción.	IEEE Std 1175.1-2002 es una guía a la familia de estándares IEEE 1175. Describe como estos estándares están pensados para ser usados para lograr la integración efectiva de herramientas de sistemas de cómputo dentro de un ambiente de ingeniería productivo y establece cuatro conceptos fundamentales en los que estos estándares están basados. Estos conceptos establecen el marco integrado para los otros miembros de esta familia de estándares. IEEE Std 1175.1-2002 describe el alcance de aplicación de cada estándar miembro, los diferentes problemas dirigidos en ese estándar, y la interrelación entre los miembros de la familia.
IEEE Std 1998	1320.1™- Estándar IEEE para el Lenguaje de Modelado Funcional. Sintaxis y Semántica para IDEF0	La función de modelado IDEF0 está diseñada para representar las decisiones, acciones y actividades de una organización existente o prospecto, ó sistema. Las gráficas y textos de acompañamiento son presentados en una forma organizada y sistemática para ganar entendimiento, análisis de apoyo, lógica para cambios potenciales, requerimientos específicos, soporte a nivel sistema de diseño y de integración de actividades. IDEF0 puede ser usada para modelar una amplia variedad de sistemas, compuestos de personas, máquinas, materiales, computadoras, y de información de todas las variedades y estructuradas por las relaciones entre ellas; tanto con las automatizadas, como con las que no.
IEEE Std 1998	1320.2™- Estándar IEEE para la sintaxis y la semántica del Lenguaje de Modelado Conceptual para IDEF1X ₉₇ (IDEF _{object})	IDEF1X ₉₇ consiste de dos lenguajes de modelado conceptual. El estilo principal que soporta el lenguaje de modelado de información/datos y es compatible de forma descendente con el estándar 1993 del gobierno de EU, FIPS PUB 184. El lenguaje de Identidad de estilos está basado sobre el modelo objeto con reglas declarativas y restricciones. IDEF1X ₉₇ identifica estilos y incluye las estructuras para los distintos pero relacionados componentes objeto de

			abstracción: Interfaces, peticiones, y realización; utiliza gráficos para establecer la interfase; y define una declarativa, Lenguaje de restricciones y reglas directamente ejecutables para la realización de peticiones.
IEEE Std 1420.1™-1995	Estándar IEEE para Tecnologías de Información -Reuso de Software- Modelo de Datos para la Interoperabilidad de Biblioteca de Reuso: Modelo de Datos de Interoperabilidad Básico. (BIDM)	proporciona el conjunto mínimo de información acerca de los recursos que la biblioteca de reuso debería poder intercambiar para la interoperabilidad.	
IEEE Std 1420.1a™-1996	Suplemento del Estándar IEEE para Tecnologías de Información -Reuso de Software- Modelo de Datos para la Interoperabilidad de Biblioteca de Reuso: Marco de Certificación de Recursos	Se define una estructura consistente para describir una política de certificación de recursos de la biblioteca de reuso en términos de un Marco de Certificación de Recursos, junto con un estándar de interoperabilidad de modelo de datos para el intercambio de información de certificación de recursos.	
IEEE Std 1420.1b™-1999	Suplemento de Uso de Ensayo del Estándar IEEE Interoperabilidad Básico (IEEE Std 1420.1-1995) para tecnologías de información -Reuso de Software- Modelo de Datos para Interoperabilidad de Biblioteca de Recursos: Marco de Derechos de Propiedad Intelectual.	Esta extensión al Modelo de Datos de Interoperabilidad Básico (IEEE Std 1420.1-1995) incorpora temas de derechos de propiedad intelectual dentro de descripciones de recurso de software para la interoperabilidad de la biblioteca de recursos.	
IEEE Std 1462™-1998	Estándar IEEE Adopción del Estándar Internacional ISO/IEC 14102:1995	ISO/IEC 14102:1995 – ISO/IEC 14102:1995 deals with the evaluation and selection of CASE tools, covering a partial or full portion of the software engineering life cycle. The adoption of the International Standard by IEEE includes an implementation note, which explains terminology differences, identifies related IEEE standards, and provides interpretation of the International Standard.	
IEEE Std 1471™-2000	IEEE Prácticas recomendadas para la descripción arquitectónica de sistemas intensivos de software	Nota: Se encuentra descrito en el área de conocimiento de Diseño de Software.	

7.2.10 Área de conocimiento de Calidad de Software.

Esta área de conocimiento trata con las consideraciones de calidad que trasciende a los procesos del ciclo de vida. Desde que la calidad del software es una preocupación presente en la ingeniería del software, también es considerada en muchas de las otras áreas de conocimiento. La descripción de áreas de conocimiento cubre cuatro subáreas.

La primer subárea describe los conceptos de calidad de software como una medida de valor de calidad, la descripción de calidad ISO 9126, fiabilidad, y otros tipos especiales de sistemas y necesidades de calidad.

La segunda subárea cubre el propósito y planeación de aseguramiento de calidad de software (SQA) y verificación y validación (V&V). Incluye actividades de planeación comunes, y planes tanto de SQA como de V&V.

La tercer subárea describe las actividades y técnicas para SQA y V&V. Incluye técnicas estáticas y dinámicas como otras pruebas de SQA y V&V.

La cuarta subárea describe mediciones aplicadas a SQA y V&V. Incluye técnicas fundamentales de medición, mediciones, análisis de medición, caracterización de defectos, y usos adicionales de datos SQA y V&V.

Tabla no. 14. Listado de Estándares que corresponden al área de Calidad de Software

Estándar	Título	Resumen
IEEE Std 730™-2002	Estándar IEEE para Planes de aseguramiento de la calidad del software	Nota: Se encuentra descrito en el área de conocimiento de Administración de la Configuración de Software.
IEEE Std 829™-1998	Estándar IEEE para la documentación de prueba de software	Nota: Se encuentra descrito en el área de conocimiento de Construcción de Software.
IEEE Std 982.1™-1988	Estándar IEEE Diccionario de medidas para producir software reutilizable.	Nota: Se encuentra descrito en el área de conocimiento de Diseño de Software.
IEEE Std 1008™-1987 (R1993)	Estándar IEEE para Pruebas de unidad de Software	Nota: Se encuentra descrito en el área de conocimiento de Construcción de Software.
IEEE Std 1012™-1998	Estándar IEEE para Verificación y Validación de Software	Se describen los procesos de Verificación y Validación de software (V&V) los cuales determinan si los productos de desarrollo de una actividad dada conforman los requerimientos de esa actividad, y si el software satisface su uso deseado y las necesidades de los usuarios. Esta determinación puede incluir análisis, evaluación, revisión, inspección, cálculo y pruebas de software y procesos. Los procesos de valoración de software V&V en el contexto de sistemas, incluyen el ambiente operacional, interfase de hardware, software, operadores, y usuarios.
IEEE Std 1012a™-1998	Suplemento del estándar IEEE para Verificación y Validación de Software: Mapa Contenido de IEEE/EIA 12207.1-1997	La relación entre los dos conjuntos de requerimientos en planes para verificación y validación de software, encontrado en el estándar IEEE 1012.1-1997, es explicada para que los usuarios puedan producir documentos que complementen a ambos estándares.
IEEE Std 1028™-1997	Estándar IEEE para	Nota: Se encuentra descrito en el área de conocimiento de Diseño de Software.

	revisiones de software.		
IEEE Std 1044™-1993	Estándar Clasificación Anomalías de software	IEEE para	Nota: Se encuentra descrito en el área de conocimiento de Prueba de Software.
IEEE Std 1061™-1998	Estándar Metodologías de Métricas de Calidad de Software	IEEE para	Nota: Se encuentra descrito en el área de conocimiento de Construcción de Software.
IEEE Std 1063™-2001	Estándar Documentación usuarios de Software	IEEE para la de	Nota: Se encuentra descrito en el área de conocimiento de Construcción de Software.
IEEE Std 1228™-1994	Estándar de la IEEE los planes de seguridad del software	para	Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de Software.
IEEE Std 1465™-1998	Estándar del estándar internacional ISO/IEC 12119:1994(E) Tecnologías de Información para requerimientos y pruebas de Calidad en Paquetes de Software.	IEEE. Adopción	Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de Software.
IEEE Std 14143.1™-2000	Adopción IEEE de ISO/IEC 14143-1:1998 Tecnologías de Información para la Medición del software, Medición del tamaño funcional. Parte 1: Definición de Conceptos		Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de Software.

7.2.11 Área de conocimiento de Ingeniería de Sistemas de Software.

La Guía del Cuerpo de Conocimiento de Ingeniería de Software no incluye un área de conocimiento para la ingeniería de sistemas. Esto es debido a que la guía SWEBOK se enfoca sobre el conocimiento específico para la ingeniería de software y deja temas relacionados, como ingeniería de sistemas, para otros esfuerzos. Es más, el Concilio Internacional en Ingeniería en Sistemas (INCOSE) está trabajando actualmente sobre su propio proyecto cuerpo de conocimiento.

Los productos de tecnología de información modernos, inclusive el intensivo software, son bastante complejos para requerir aplicaciones de técnicas desde ambas disciplinas. Andriole y Freeman escriben que “han encontrado que la creación de sistemas intensivos de software demandan más que lo tradicional en cualquier campo”. Como consecuencia, ha llegado a ser importante para entender la relación entre sistemas y la ingeniería de software.

Existen muchas definiciones de ingeniería de sistemas. La definición desde IEEE 100 es “La aplicación de las ciencias matemáticas y física para desarrollar sistemas que utilizan económicamente los materiales y fuerzas de la naturaleza para el beneficio de la humanidad”.

Desde este punto de vista, muchas de las disciplinas de la ingeniería de software han sido heredadas desde la ingeniería de software. Aunque esta colección de estándares se enfoca en la ingeniería en sistemas, se debe recordar que la ingeniería en sistemas aplicará disciplinas similares en un amplio sentido a los requerimientos de todos los sistemas.

Tabla no. 15. Listado de Estándares que corresponden al área de Ingeniería de Sistemas de Software

Estándar	Título	Resumen
IEEE Std 1220™-1998	Estándar IEEE para la Aplicación y Mantenimiento de los Procesos de Ingeniería de Sistemas	Se definen las tareas interdisciplinarias, las cuales son requeridas durante todo el ciclo de vida de los sistemas para transformar las necesidades de los clientes, requerimientos, y restricciones dentro de soluciones de sistemas. Además, los requerimientos para los procesos de ingeniería en sistemas y sus aplicaciones desde los productos del ciclo de vida especificados. La importancia de este estándar está en las actividades de ingeniería necesarias para guiar desarrollo de productos mientras se asegura que el producto está propiamente diseñado para hacerlo asequible para producir, poseer, operar, mantener y eventualmente desecharlo, sin riesgos indebidos a la salud o al ambiente.
IEEE Std 1228™-1994	Estándar de la IEEE para los planes de seguridad del software	Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de Software.
IEEE Std 1233™, 1998 Edition	Guía de la IEEE para las Especificaciones de Requerimientos en el Desarrollo de Sistemas	Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de Software.
IEEE Std 1362™-1998	Guía IEEE para Definición de conceptos de Sistemas de Tecnologías de Información, de documentos de operación (ConOps)	Nota: Se encuentra descrito en el área de conocimiento de Requerimientos de Software.
IEEE Std 1471™-2000	IEEE Prácticas recomendadas para la descripción arquitectónica de sistemas intensivos de software	Nota: Se encuentra descrito en el área de conocimiento de Diseño de Software.

7.2.12 Área de conocimiento de Mejores Prácticas de Internet

La guía del cuerpo de conocimiento de la ingeniería de software se enfoca en la aproximación a la ingeniería de software más que en sus resultados. Las disciplinas maduras de la ingeniería, sin embargo, codifican las soluciones para clasificar los problemas. Los ingenieros practicantes consultan los diseños de sus predecesores antes de intentar diseños novedosos. La colección SESC incluye un documento que falla dentro de esta categoría. El estándar sobre las mejores prácticas de internet aplica los principios de la ingeniería de software para proveer una guía específica para solucionar una clase de producto de ingeniería de software.

La internet se está expandiendo y su valor se está incrementando como un método para localizar y entregar información. Esto significa cambios en la ingeniería. La localización aplicable a la información requiere de información indexada que sea incorporada dentro del desarrollo de páginas Web. Una vez que una página aplicable ha sido localizada, la información esencial puede no estar presente, resultando en una frustración para el usuario y una falla de la aplicación Web para cumplir su propósito.

Seguido la ingeniería de páginas Web está hecha con pequeñas consideraciones inmediatas o implicaciones continuas del diseño o implementación de sitios Web. Algunos sitios reflejan que la entrega “el estado del arte” puede solo ser accesada con las herramientas más recientes. Lo que puede ser inconsistente con los objetivos de los negocios para ese sitio. Algunos sitios languidecerán antes de su vida aplicable, ocupando recursos valiosos. Una ingeniería de páginas Web pobre resulta en una pobre productividad y frustración de usuarios, y puede repercutir en obligaciones legales.

El propósito del estándar para mejores prácticas es el de reducir los riesgos asociados con las inversiones en páginas Web.

Tabla no. 16. Listado de Estándares que corresponden al área de Mejores Prácticas de Internet

Estándar	Título	Resumen
IEEE Std 2001™-2002	Práctica recomendada IEEE para Prácticas en Internet –Ingeniería de Páginas Web- Aplicaciones Extranet/Intranet	Se definen las prácticas recomendadas para la ingeniería de páginas World Wide Web para ambientes de Intranet y Extranet basadas en el Consorcio World Wide Web (W3C) y las guías de industrias relacionadas. Estas prácticas no dirigen consideraciones de estilo o consideraciones de factor humano en el diseño de páginas Web más allá de las limitaciones que reflejan la buena práctica de ingeniería.

7.3 Estructuras Organizacionales.

Según Prosoft, existen en el país aproximadamente 390 empresas desarrolladoras de software. Sin embargo, el tamaño de la demanda internacional es de tal magnitud, que favorece la creación de una gran cantidad de nuevas empresas.

Según [Pérez, 2005] los programas de mejora de procesos de desarrollo de software en México son relativamente nuevos. Hasta hace algunos años el éxito de los proyectos se fundamentaba principalmente en la capacidad de sus directivos y en el personal que laboraba en los proyectos de las diferentes organizaciones. En la actualidad son los propios directivos quienes se dan cuenta que esa capacidad no es suficiente y que es necesario definir procesos y establecer la disciplina que permita repetir y reutilizar las experiencias obtenidas de proyectos exitosos. En menos de diez años este enfoque ha tomado presencia dentro de las organizaciones de desarrollo de software y áreas internas de TI. Cada vez es más demandado y reconocido el éxito de los proyectos respaldados por un enfoque de procesos. La capacidad que se logra permite llevar de un

desarrollo artesanal a un desarrollo industrial considerado. Pero aún queda mucho por hacer, el alcance de estos programas no rebasa 5% del total de las organizaciones de desarrollo de software en México.

Pérez presenta además el tipo de servicio que prestan las empresas de desarrollo de software y su proporción (ver figura 9).

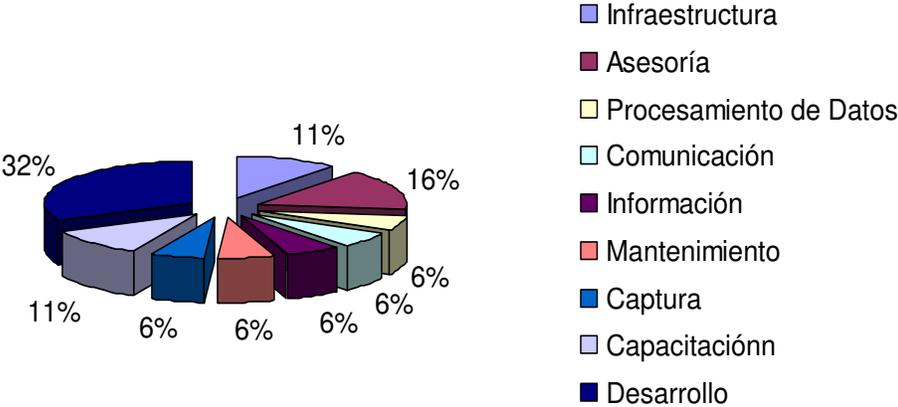


Figura no. 9. Proporción de servicios que prestan las empresas

Según los resultados anteriores, el desarrollo de software es el servicio más prestado por las miPymes de software en México, por lo que el propósito principal de esta investigación es el de proporcionar a las miPymes de software una fuente de información para determinar los estándares de la IEEE que satisfagan sus necesidades en el desarrollo de software.

CAPITULO VIII DESARROLLO

A continuación se presenta el desarrollo de la investigación llevada a cabo, en donde se seleccionó de las MiPymes; el tipo de empresa a estudiar y del conjunto de estándares de calidad de software de la IEEE, aquellos que se utilizaron para su estudio. Además, se presentan dos formas de aplicar los estándares elegidos: De acuerdo al ciclo de vida en cascada, y de acuerdo a los proyectos a desarrollar.

8.1 Selección del tipo empresa a estudiar

Como se vió en el tema 2.2.1, el porcentaje de micro empresas de software registradas por la secretaría de economía en México es del 90.79% del total de las empresas. Además, a pesar de ser mayoría, según Santillán [evento prosoft noviembre, 2005], más del 90% de las microempresas no llegan a cumplir cinco años debido a que no cuentan entre otras cosas, con procesos de calidad. Este dato es preocupante, ya que se cuenta con mano de obra competitiva, actualmente el gobierno federal se encuentra involucrado para apoyar a esta industria junto con cámaras y asociaciones.

Añadió además, que en los próximos cuatro años estas microempresas, generarán 350,000 nuevos empleos (considerando áreas como de recursos humanos, ventas, etc.) y dos billones de nuevos impuestos. Por lo que para mejorar la industria de software, se deben balancear los recursos empleando la siguiente estrategia:

Calidad + Competitividad

La táctica que el gobierno federal está utilizando, es la promoción de estándares que les ayuden a alcanzar esa meta.

Es por esta razón que la presente investigación se enfocará en la microempresa, seleccionando y describiendo los estándares de la IEEE adecuados a sus características específicas.

La microempresa en donde se aplicarán los estándares es Óptima Computación, con domicilio en la calle Hnos. Aldama 234-A, colonia centro en Celaya, Gto (ver figura no. 10). Su giro es el de servicios informáticos y venta de equipo de cómputo.

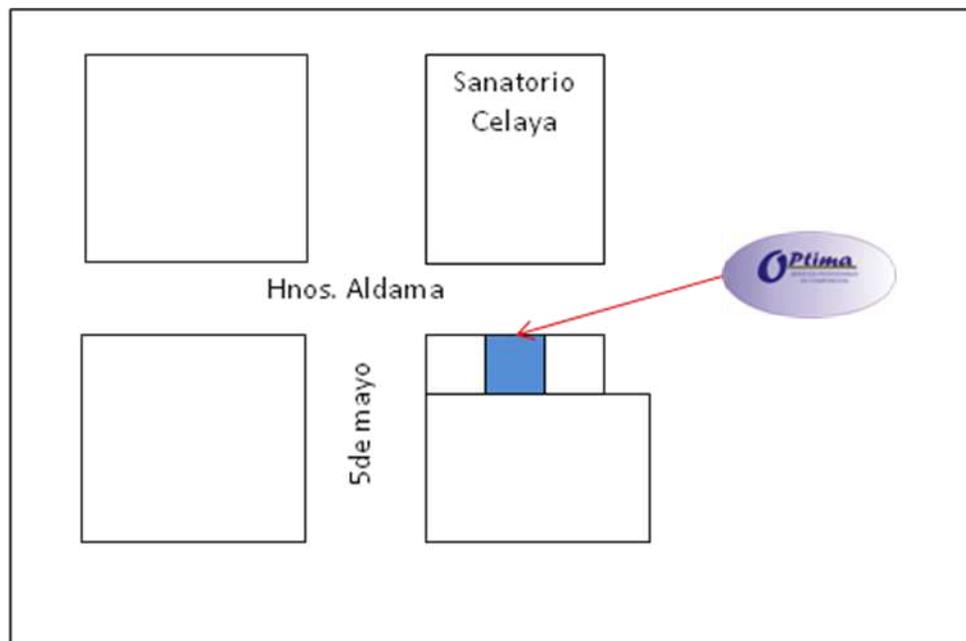


Figura no. 10. Ubicación de la microempresa.

La microempresa se creó en el mes de junio del año 2000 y cuenta con 5 empleados. El gerente, un Ingeniero Electrónico para el área de mantenimiento, 2 Ingenieros en Sistemas para el Área de Desarrollo, 1 Ingeniero en Sistemas para el área de Ventas y una secretaria (ver figura no. 11). Las actividades contables propias de la empresa, las lleva un Contador externo.

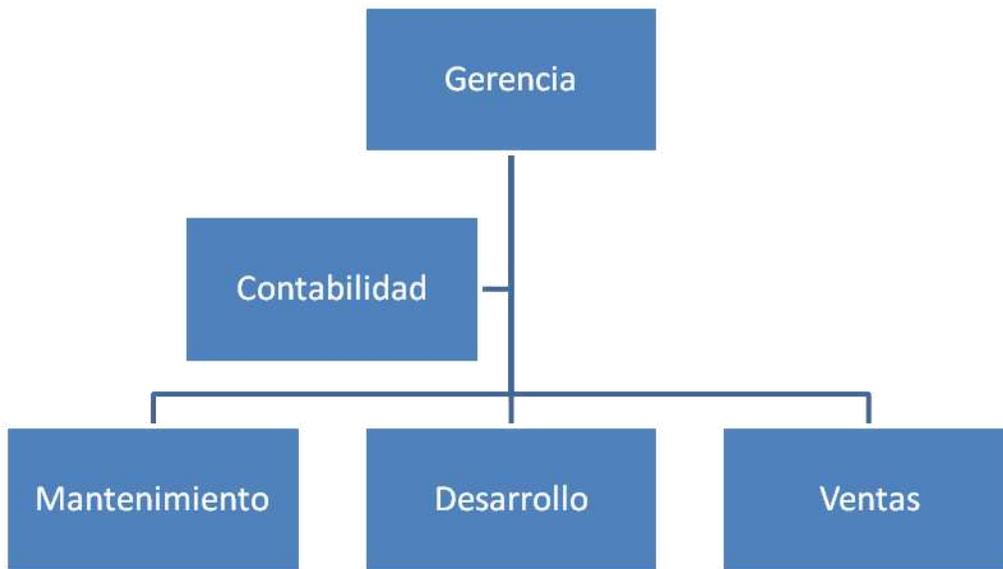


Figura no. 11. Ubicación de la microempresa.

La microempresa inició únicamente con el servicio de venta y mantenimiento de equipo de cómputo y con un ingeniero Electrónico, además de un Ingeniero en Sistemas Computacionales. A partir del 2002, Óptima Computación crece comenzando a ofrecer sus servicios de desarrollo de software, teniendo como su primer cliente al municipio de Celaya, específicamente la Contraloría municipal.

Para el primer proyecto que se llevó a cabo en Contraloría, le microempresa no tenía el conocimiento de la existencia de los estándares de calidad de software de la IEEE y se presentaron varios problemas en las diferentes etapas del ciclo de vida del proyecto, de tal forma que de el tiempo que se tenía planeado para terminar el sistema que era de seis meses, realmente se utilizaron 11 meses.

De entre los problemas que se tuvieron en el desarrollo del proyecto, se tuvieron errores graves en la Determinación de requerimientos y en el Análisis de

Sistemas, de tal forma que para casi la entrega del proyecto, hubo que rediseñar la base de datos, pantallas y por consiguiente el código de algunos módulos.

El conjunto de estándares de calidad de software de la IEEE, son creados para apoyar a los desarrolladores de software en las diferentes etapas del ciclo de vida de los sistemas, de tal forma que la microempresa Óptima Computación buscará implementarlos para eliminar sus errores, buscando la mejora continua.

8.2 Selección del sector del área de conocimiento a estudiar.

Según García [2005], existen diversos modelos de procesos de software, que definen procesos claves y actividades a realizar dentro de las organizaciones. Entre los más conocidos están PSP, TSP, CMMI, IEEE 12207. Sin embargo, las microempresas están poco familiarizadas con estándares de procesos. El problema que surge para que puedan considerarlos en sus prácticas, es la falta de infraestructura para soportarlas.

Considerando el estándar IEEE 12207, se propone una estructura de 5 procesos primarios (enfocados a la razón de ser de una empresa) y procesos de soporte enfocados a la calidad del proceso, y para garantizar la calidad de producto; tiene la ventaja de ser muy general ya que con ello las empresas lo pueden adaptar de acuerdo a la estructura organizacional y a los recursos con los que cuentan.

Debido a lo anterior, se seleccionarán del conjunto de estándares de software de la IEEE, aquellos que estén enfocados al desarrollo de software, los cuales se organizarán de acuerdo al ciclo de vida del software.

8.3 Selección de estándares

Para la micromepresa Óptima Computación, se llevó a cabo la selección de los estándares que sirvieran para el presente estudio. Se tomó en cuenta a aquellos que se relacionen con el desarrollo ciclo de vida del software y que además puedan ser aplicados a la microempresa, ya que son los procesos que se encuentran con mayor frecuencia en ellas.

Tabla no. 17. Selección de estándares.

Clave	Estándar	Título	
A	830	Prácticas Recomendadas por la IEEE para la Especificación de Requerimientos de Software	
B	1016	IEEE	Prácticas recomendadas para las descripciones de diseño de software
C	1028	Estándar	IEEE para revisiones de software
D	1008	Estándar	IEEE para Pruebas de unidad de Software
E	1219	Estándar	IEEE para mantenimiento de software

8.4 Actividades a realizar de acuerdo al ciclo de vida de cascada

Según Bruegge [2002], un modelo del ciclo de vida del software representa todas las actividades y productos de trabajo necesarios para desarrollar un sistema de software. Los modelos de ciclo de vida permiten que los

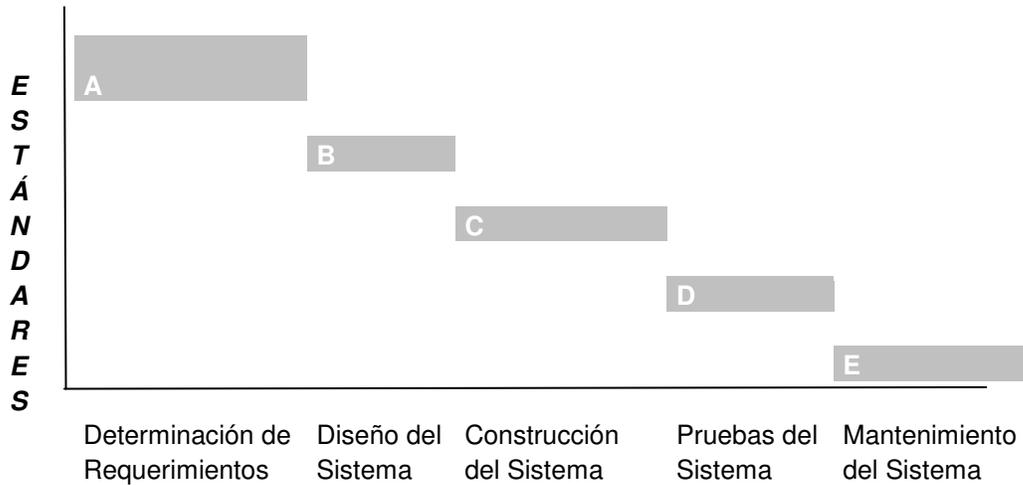
gerentes y desarrolladores manejen la complejidad del proceso de desarrollo de software en la misma forma que un modelo de análisis o un modelo de diseño del sistema permite que los desarrolladores manejen la complejidad de un sistema de software.

Para el presente estudio, se eligió el ciclo de vida de cascada, ya que es el utilizado por la mayoría de las microempresas, por lo que facilitará a las estas, la adecuación de los estándares a sus procesos.

El modelo de cascada, añade Bruegge [2002], es un modelo del ciclo de vida centrado en actividad que prescribe una ejecución secuencial de un subconjunto de los procesos de desarrollo y administrativos. Su característica principal es que cada actividad de desarrollo deberá ser seguida por una revisión. El punto inicial de este modelo es la actividad de análisis de requerimientos del sistema, cuyo objetivo es generar requerimientos del sistema no ambiguos. La primera revisión será la Revisión de los Requerimientos del Sistema, durante la cual se revisan la suficiencia, consistencia y claridad de los requerimientos. Los requerimientos del sistema son la base para la actividad del diseño del sistema. El diseño del sistema se revisa durante la actividad de Revisión del diseño del sistema.

El diseño del sistema se toma como base una vez que se termina en forma satisfactoria su revisión. La implementación se inicia con el diseño preliminar, seguido por la actividad de diseño detallado. Una revisión importante es la revisión del diseño crítico. La codificación no empieza antes de que la revisión a este diseño se termine en forma satisfactoria.

En la figura no. 10, se muestran las actividades el ciclo de vida de cascada llevadas a cabo por las microempresas de desarrollo de software y se incluyen los estándares de software de la IEEE que se deben llevar en cada actividad.



ACTIVIDADES DEL CICLO DE VIDA

Figura no. 10. Organización de estándares de acuerdo al ciclo de vida

8.5 Actividades a realizar de acuerdo al tiempo

Para asignar los estándares a las actividades de acuerdo a los tiempos, se establecieron cuatro etapas las cuales se aplicaron a diferentes proyectos de una microempresa de software. Las cuatro etapas son:

- a) Análisis del estado actual
- b) Implantación del estándar
- c) Maduración
- d) Mejora Continua

8.5.1 Análisis del estado actual.

En esta etapa se mencionará la situación actual de la microempresa en estudio:

- Utilizan siempre el ciclo de vida en cascada para todos sus desarrollos de software.
- Existe una falta de organización en su historial de proyectos, pruebas, documentación, estandarizaciones y evaluaciones de procesos.

- Planifican y realizan presupuestos de sus proyectos, pero no coinciden con estimaciones reales.
- En el 80% de sus proyectos, en la actividad de construcción del sistema se solicita nuevamente la actividad de determinación de requerimientos.
- Se tiene una alta dependencia de la persona con más conocimientos en programación.
- Si hay plazos rígidos, se sacrifican funcionalidad y calidad del producto para satisfacer el plan
- No existen bases objetivas para juzgar la calidad del producto
- Cuando los proyectos está fuera de plan, las revisiones o pruebas se recortan o eliminan

La situación anteriormente descrita de la microempresa en cuestión es semejante a la de la mayoría de las microempresas desarrolladoras de software. Es por esta razón por la cual se tomó en cuenta a esta microempresa para el presente estudio.

8.5.2 Implantación del estándar

La empresa en cuestión estudió a tres proyectos diferentes en tiempos diferentes a los cuales se les implantó el estándar de la IEEE 830: Prácticas recomendadas por la IEEE para le especificación de requerimientos. En la figura no. 11 se muestra la gráfica en donde se propone la implantación de los diferentes estándares por proyecto en relación con el tiempo.

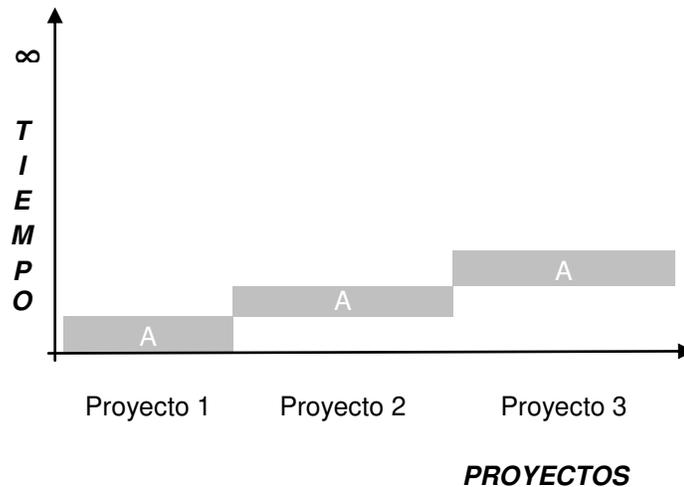


Figura no. 11. Implantación del estándar A por proyecto.

En el primer proyecto en investigación, la empresa no contaba con algún método de medición de la calidad, por lo que se implantó el estándar clasificado como A, para la obtención del documento de requerimientos. Esta implantación se realizó en conjunto con una persona de la microempresa seleccionada para aprender la administración del estándar (Ver Anexo I).

La implementación del estándar se llevó a cabo en los cuatro primeros meses del proyecto y la duración total del proyecto fue de un año. Para el segundo proyecto, se volvió a implementar tomándose en cuenta la experiencia y el documento del proyecto anterior, además de que la persona elegida fue la encargada de verificar el desarrollo del estándar.

La implementación del estándar se llevó a cabo en un mes y medio y la duración total del proyecto fue de siete meses. Se utilizó un tercer proyecto para probar la experiencia del encargado de verificar los procesos.

El proyecto se terminó en un año y la implementación del estándar se abarcó el primer mes y medio de la duración total del proyecto. El sistema se pudo verificar contra el documento de requerimientos, en donde se comprobó que el

sistema se terminó en tiempo y forma como se había predicho en el documento de requerimientos.

Hasta el tercer proyecto se tuvo la oportunidad de hacer la presente investigación. A través de estos proyectos se fue observando la experiencia que fue ganando la empresa y para el cuarto proyecto, se puede afirmar que se contará con los procesos de requerimientos documentados, definidos, practicados; por lo que ya se puede hablar de que en cuanto al estándar clasificado como A, la empresa está en un nivel de mejora continua, y se propone que en ese momento, se puede comenzar la implementación del siguiente estándar clasificado como B, 1016, IEEE Prácticas recomendadas para las descripciones de diseño de software.

De acuerdo a la investigación realizada a la empresa en cuestión, se propone a las miPymes de software la implementación de los estándares seleccionados en el punto 8.3, como se muestra en la figura no. 12. Por lo que la terminación de la implementación total de los estándares seleccionados será aproximadamente en 16 proyectos.

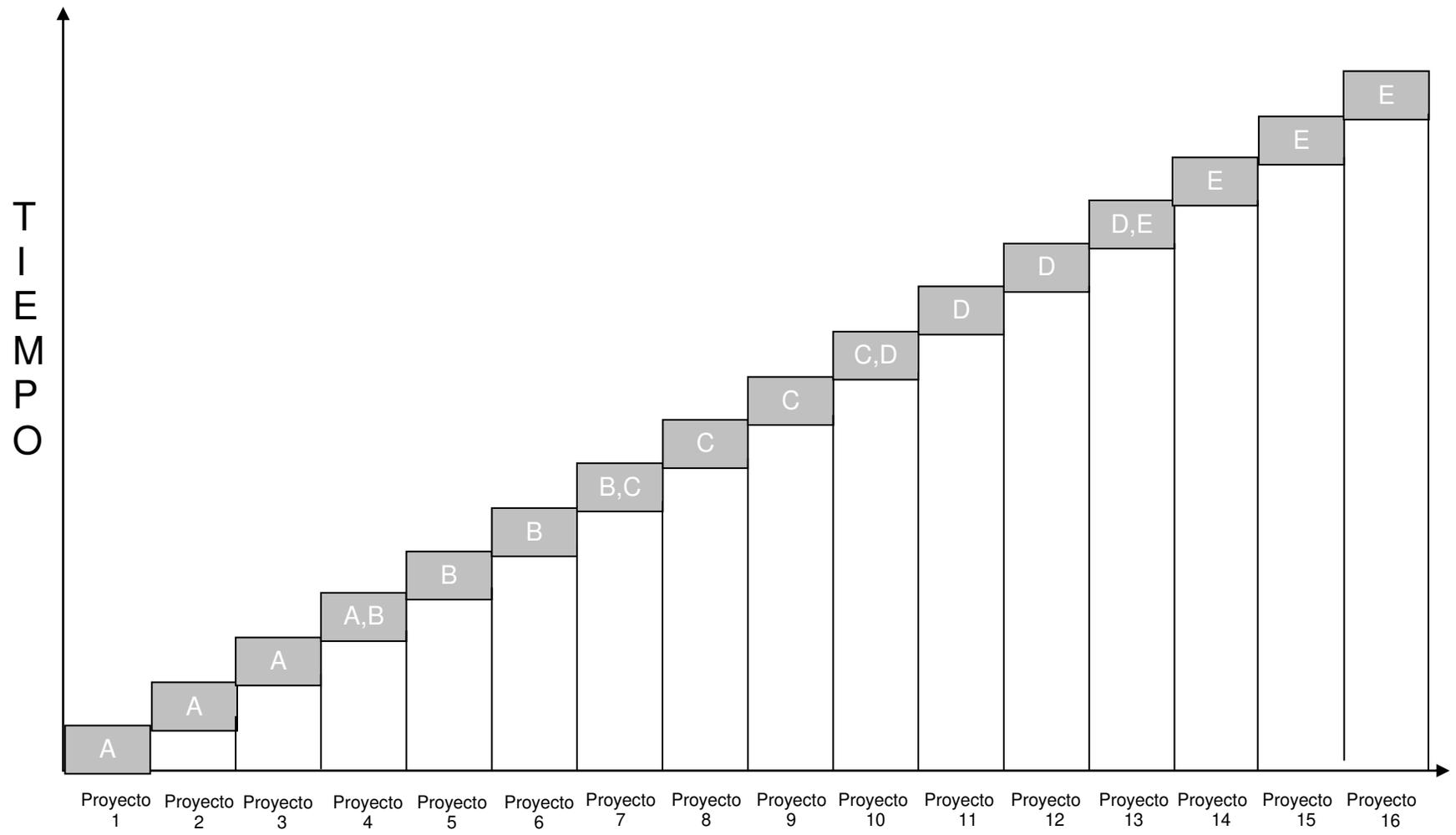


Figura no. 12. Propuesta de implementación de los estándares seleccionados por proyecto.

La presente investigación propone que para el cuarto proyecto (figura no. 12), los procesos para la Determinación de Requerimientos del estándar clasificado como A, deberán contener las características de un proceso maduro (el cual se tratará en el siguiente tema), por lo que se puede continuar con la aplicación del siguiente estándar clasificado como B y así sucesivamente.

8.5.3 Maduración

Después de presentar la propuesta de la implementación de los estándares, a continuación se deberá determinar el grado de madurez que alcancen los procesos en caso de que las miPyme de desarrollo de software implementen los estándares, para posteriormente implementar medidas que garanticen la mejora continua de estos.

La maduración de un proceso se define como el nivel al cual el proceso está explícitamente documentado, gestionado, medido, controlado y continuamente mejorado. De acuerdo a lo anterior, se elegirá a una persona de la microempresa para que sea la encargada de llevar a cabo la verificación de los procesos para el desarrollo de software que se realicen a través de los proyectos en donde se implanten los estándares para determinar el grado de madurez de los procesos (ver figura no. 13).

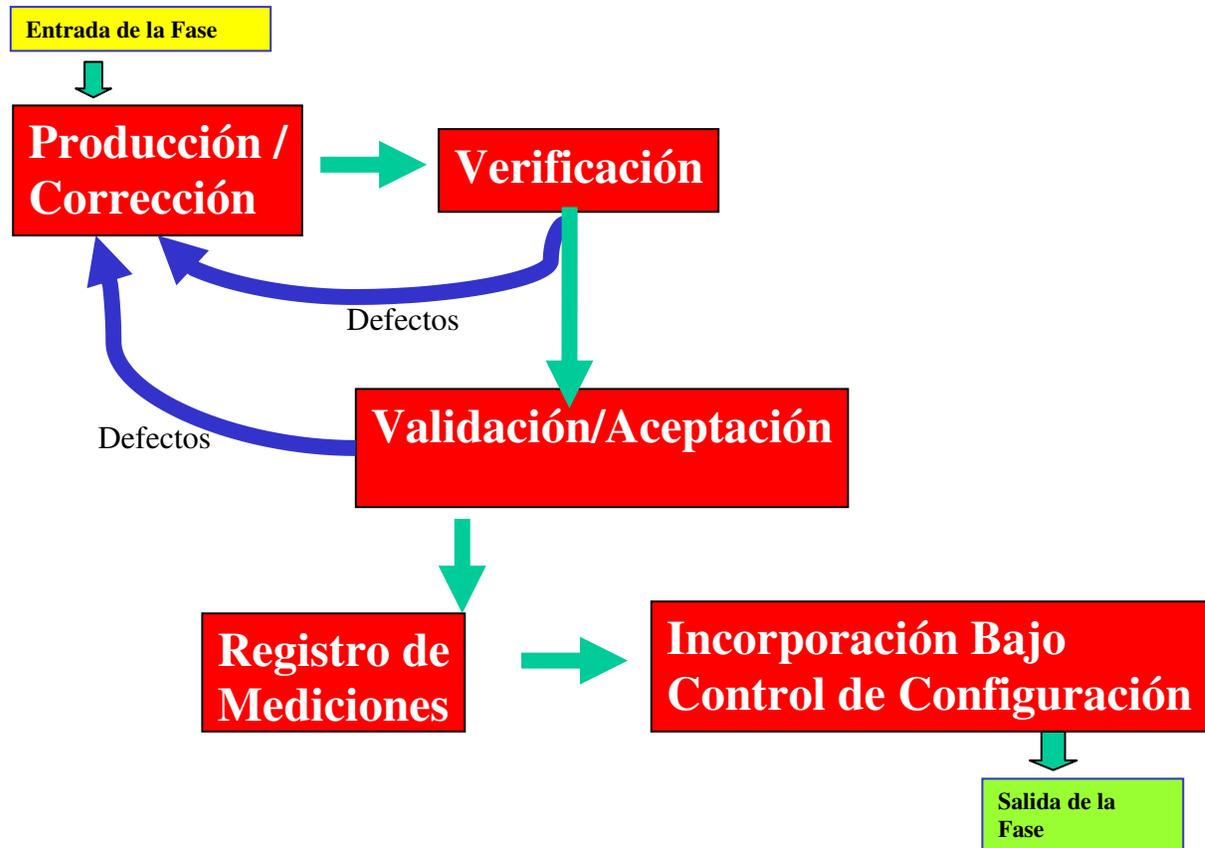


Figura no. 13. Fase de verificación de procesos

La evaluación se apoya en el modelo de maduración de capacidades (CMM) el cual, según Bruegge [2002] asume que el desarrollo de los sistemas de software se hace más predecible cuando una organización usa un proceso de ciclo de vida bien estructurado, visible para todos los participantes en el proyecto y que se adapta al cambio. El CMM usa los siguientes cinco niveles para caracterizar la madurez de una organización:

- a) Nivel 1: Inicial. En donde el desempeño es basado en la competencia del personal
- b) Nivel 2: Repetible. En este nivel cada proyecto tiene un modelo de ciclo de vida bien definido. Sin embargo la organización:

- c) Nivel 3: Definido. Este nivel usa un modelo de software documentado para todas las actividades administrativas y técnicas para toda la organización:
- d) Nivel 4: Administrado. Este nivel define las medidas para las actividades y los productos a entregar. En este momento la organización:
- e) Nivel 5: Optimizado. Los datos de las mediciones se usan en un mecanismo de retroalimentación para mejorar el modelo de ciclo de vida del software a lo largo de la vida de la organización e identifica y elimina causas de desempeño pobre (ver figura no. 14).

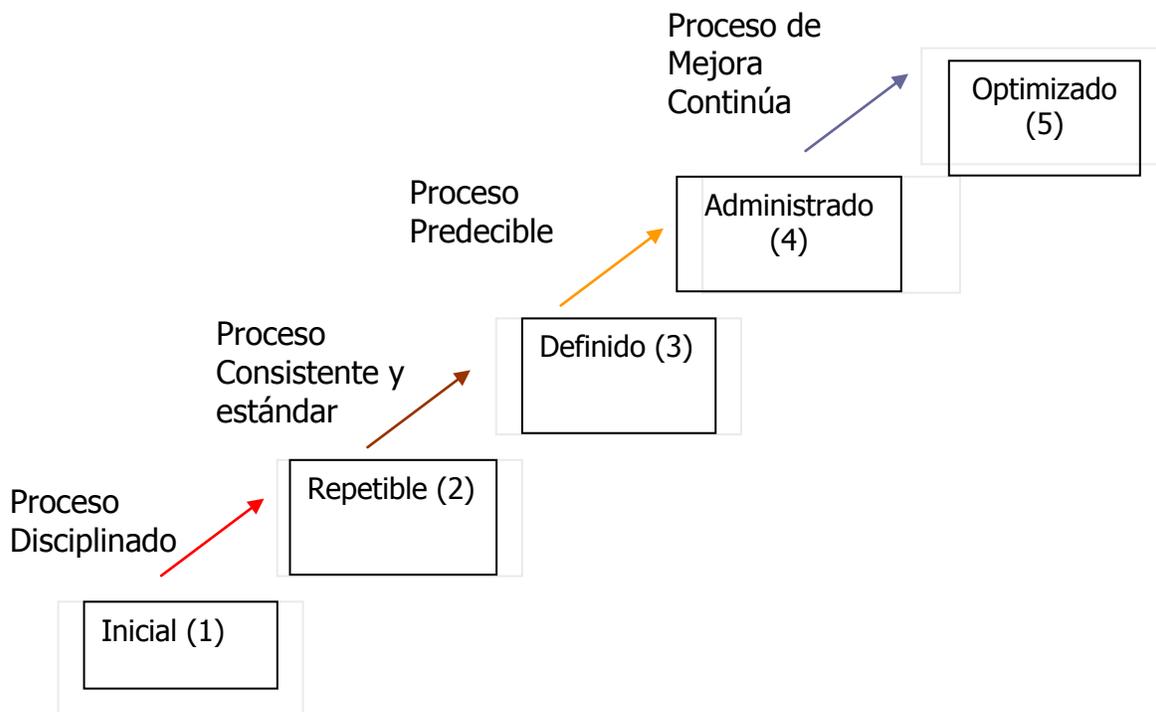


Figura no. 14. Niveles de Madurez en los procesos del Software

Para llevar a cabo la evaluación de los procesos y determinar así el nivel de madurez de la empresa, CMM evalúa las áreas de proceso claves (KPA's) de acuerdo a cada nivel, como lo muestra la tabla no. 18:

Tabla no. 18. Áreas de proceso claves para CMM

Nivel de Madurez	Área de proceso clave
Inicial	No es aplicable
Repetible	Administración de Requerimientos Planificación de Proyecto de Software Seguimiento y Supervisión de proyectos de Software Administración de Subcontratos de Software Aseguramiento de la calidad del Software Administración de la Configuración del Software
Definido	Enfoque en el proceso de la organización Definición del proceso de la organización Programa de entrenamiento Administración integrada de software Ingeniería de Productos de software Coordinación entre grupos Revisiones entre iguales
Administrado	Administración cuantitativa del proceso Administración de la calidad del software
Optimizado	Prevención de defectos Administración del cambio tecnológico Administración del cambio de procesos

Los procesos se evaluarán de la siguiente forma: N (Not), P (Partially), L (Largely) y F (Fully), siendo:

- **N** No alcanzado (0% a 15%): Poca o ninguna evidencia de la consecución del atributo
- **P** Parcialmente alcanzado (16% a 50%): Evidencia de un enfoque sistemático y de la consecución del atributo. aunque algunos aspectos de la consecución pueden ser impredecibles
- **L** Ampliamente alcanzado (51% a 85%): Evidencia de un enfoque sistemático y de una consecución significativa del atributo. La realización del proceso puede variar en algunas áreas
- **F** Totalmente alcanzado (86% a 100%): Evidencia de un enfoque completo y sistemático y de la consecución plena del atributo

La figura no. 15 presenta el diagrama que muestra las actividades principales (Planeación, Realización y Evaluación y Control) junto con su flujo

correspondiente para llevar a cabo la verificación de la implementación de los estándares y determinar el grado de maduración.

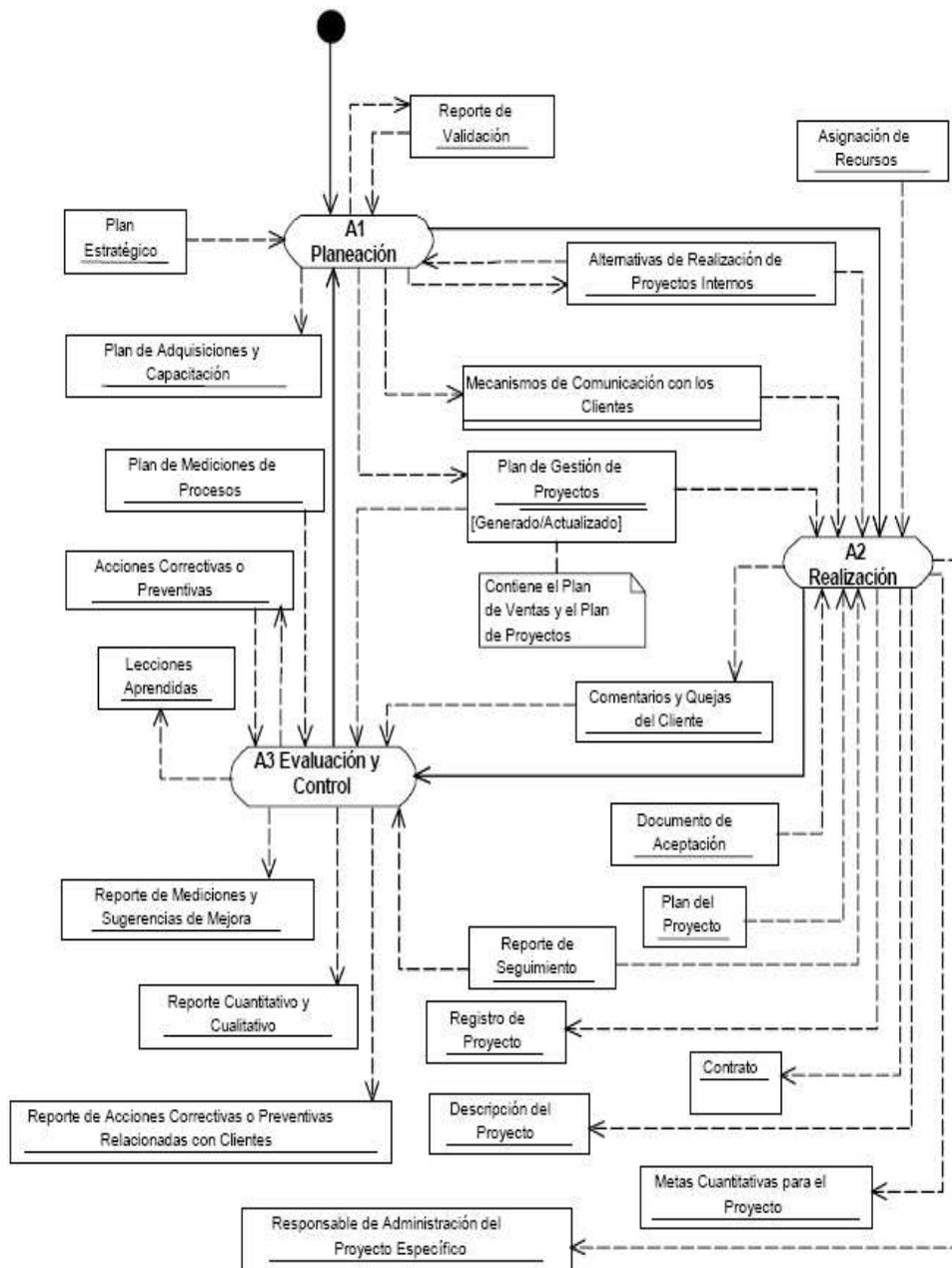


Figura no. 15. Diagrama de flujo para alcanzar la evaluación y control de los procesos

Alcanzar la fase de maduración de procesos requiere un cambio de cultura de trabajo que implicará la inversión de tiempo y esfuerzo; sin embargo las

ventajas de garantizar al cliente un producto con calidad, van más allá de mantener al cliente mismo; se trata de poner a la empresa en un grado de competitividad internacional que conlleva al crecimiento y asegura el mantenimiento de la empresa en el mercado.

Los procesos por sí solos no permanecen por tiempo indefinido en un grado de madurez, por lo que la miPyme deberá hacer uso de estrategias que garanticen la mejora continua

8.5.4 Mejora Continua

El Mejoramiento Continuo es un proceso que describe muy bien lo que es la esencia de la calidad y refleja lo que las empresas necesitan hacer si quieren ser competitivas a lo largo del tiempo. Es algo que como tal es relativamente nuevo ya que se puede evidenciar en las fechas de los conceptos emitidos, pero a pesar de su reciente natalidad en la actualidad se encuentra altamente desarrollado.

La importancia de esta técnica gerencial radica en que con su aplicación se puede contribuir a mejorar las debilidades y afianzar las fortalezas de la organización, a través de este se logra ser más productivos y competitivos en el mercado al cual pertenece la organización, por otra parte las organizaciones deben analizar los procesos utilizados, de manera tal que si existe algún inconveniente pueda mejorarse o corregirse; como resultado de la aplicación de esta técnica puede ser que las organizaciones crezcan dentro del mercado y hasta llegar a ser líderes. La base del éxito del proceso de mejoramiento es el establecimiento adecuado de una buena política de calidad, que pueda definir con precisión lo esperado por los empleados; así como también de los productos o servicios que sean brindados a los clientes.

La Mejora Continua, significa mejorar los estándares, estableciendo a su vez, estándares más altos, por lo que una vez establecido este concepto, el trabajo de mantenimiento por la administración o por el responsable del proceso, consiste en procurar que se observen los nuevos estándares.

La Mejora Continua duradera, sólo se logrará cuando el personal trabaja para estándares más altos, de este modo, el mantenimiento y el mejoramiento son una mancuerna inseparable. Por tal motivo, cuando se efectúan mejoras en los procesos, éstas a la larga, conducirán a mejorar la calidad y la productividad, evitando así, la preocupación por los resultados (ver figura no. 16)

La figura no. 17 muestra el ciclo para mantener la calidad a través de la mejora continua según Sexto [2006]. El autor señala que primer paso del ciclo es Planificar, en donde se proponen las siguientes acciones:

- Involucrar a la gente correcta que llevará a cabo el proceso de mejora.
- Recopilar los datos disponibles.
- Comprender las necesidades de los clientes
- Estudiar los procesos involucrados
- Es el proceso capaz de cumplir las necesidades?
- Desarrollar el plan, entrenar al personal

El siguiente paso dentro del ciclo será el de Hacer, el cual contiene las siguientes actividades:

- Implementar la mejora/verificación de las causas de los problemas
- Recopilar los datos apropiados

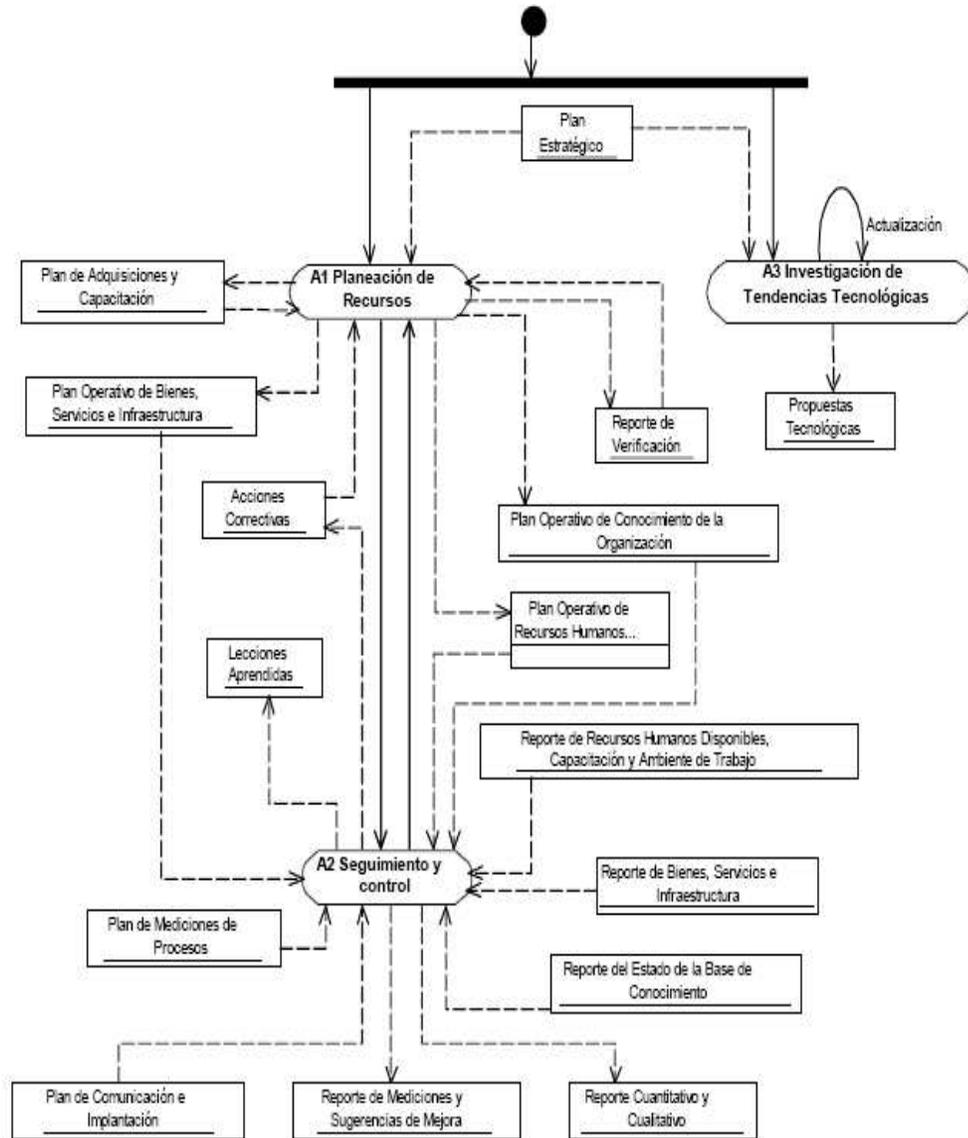


Figura no. 16. Diagrama de flujo para garantizar la mejora continua

El tercer paso será el Verificar y contempla las siguientes actividades:

- Analizar y desplegar los datos
- Se han alcanzado los resultados deseados?

- Comprender y documentar las diferencias
- Revisar los problemas y errores
- ¿Qué se aprendió?
- ¿Qué queda aún por resolver?

El cuarto y último paso del ciclo corresponde al Actuar, y contiene las actividades que a continuación se listan:

- Incorporar la mejora al proceso
- Comunicar la mejora a todos los integrantes de la empresa
- Identificar nuevos proyectos/problemas

Al finalizar el cuarto paso, se repetirá el ciclo una y otra vez.

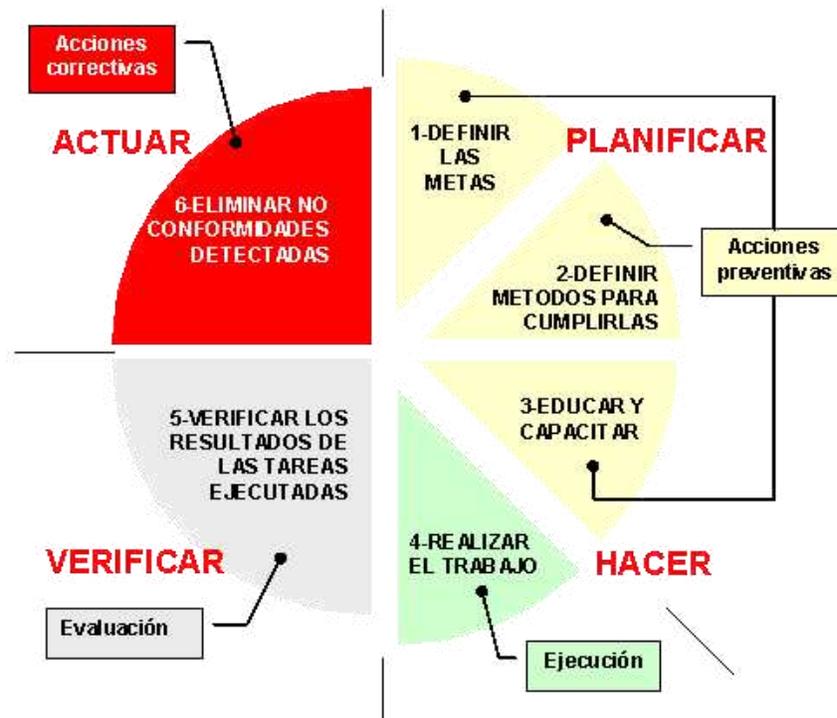


Figura no. 17. Ciclo de la calidad

Como se puede ver, los pasos del Ciclo requieren recopilar y analizar una cantidad sustancial de datos. Para cumplir el objetivo de mejora deben realizarse correctamente las mediciones necesarias y alcanzar consenso tanto en la definición de los objetivos/problemas como de los indicadores apropiados. Se dispone de una serie de herramientas para desarrollar esta tarea, son las llamadas “Herramientas de la Calidad” y la mayoría se basa en técnicas estadísticas sencillas. Algunos ejemplos son :

- Diagramas de Causa-Efecto
- Listas de Verificación
- Diagramas de Flujo
- Distribuciones de Frecuencia e Histogramas
- Diagramas de Pareto
- Gráficos de Control

PARTE III

RESULTADOS

CAPÍTULO IX RESULTADOS

La importancia de administrar y documentar los diferentes procesos utilizados en el ciclo de vida de los sistemas, debe ir más allá del interés de una certificación. Verdaderamente cada vez más empresas exigen que sus proveedores de software estén certificados, pero una microempresa debe tomar en cuenta también otros objetivos como el de mejorar los procesos para ampliar su mercado, contar con planes de crecimiento, capacitación, entre otros.

Sin embargo, a través de este estudio se encontró que las microempresas desconocen estas ventajas o están poco interesadas en ellas ya que consideran que no cuentan con los recursos económicos, humanos y de infraestructura para llevarlas a cabo. Aunque esto repercute en una mala organización y la pérdida en consecuencia de sus clientes.

Con la presente investigación, se planteó la importancia de la calidad aplicada en los procesos de desarrollo del software. Mostrando las ventajas y las necesidades de la implementación de los estándares en los procesos de desarrollo

Se investigó el conjunto de los cuarenta estándares de la IEEE para el desarrollo del software, se clasificaron y se eligió un subgrupo que se propuso para implementarlos en las miPymes de Desarrollo de Software.

Se propuso la implementación de los estándares elegidos de acuerdo al ciclo de vida de cascada y se determinó un total de 16 proyectos para la implementación completa de los estándares elegidos.

CAPÍTULO X CONCLUSIONES

Los estándares de calidad de software de la IEEE, son una herramienta que ayuda a las microempresas a alcanzar niveles de madurez en sus procesos para lograr el mejoramiento de sus procesos y alcanzar niveles de calidad.

Sin embargo las microempresas de software en México han sido reacias a incurrir en esta área y lo hacen patentes las estadísticas que muestran un promedio de vida de estas microempresas de cinco años.

Realmente existen estándares que no podrían ser implantados en estas microempresas por sus propias características, sin embargo la presente investigación nació del interés de facilitar el acceso a los estándares y seleccionar aquellos que fueran factibles de implantarse y así ayudar a alcanzar un nivel de madurez en los procesos de las microempresas de desarrollo de software.

Se seleccionó el ciclo de vida de cascada para llevar a cabo la implantación de los estándares de acuerdo a cada fase del ciclo de vida. A este tipo de implantación, se continuó haciéndolo por proyectos en un período de tiempo, en donde se propone implantar nuevos estándares en diferentes tiempos, culminando en el proyecto cuatro, en donde se podrán implantar cinco estándares de la IEEE que contemplan las diferentes fases del ciclo de vida de los sistemas.

CAPÍTULO XI TRABAJO FUTURO

Después de terminar la presente investigación y plantear la importancia del uso de estándares de calidad implementados a las miPymes de Desarrollo en México, se encontró que estas miPymes no solo desarrollan sistemas informáticos administrativos enfocados a una arquitectura cliente-servidor

Actualmente las empresas en general están haciendo uso cada vez mayor de la Tecnologías de Información y estas asu vez, están más al alcance de las posibilidades del público en general. De esta forma, el Internet es una de las Tecnologías de Información que más auge ha tenido debido a su alcance mundial y a sus precios en relación a otros medios de comunicación para promocionar los productos o servicios de las empresas, por lo que cada vez es más importante el desarrollo de sitios web.

El conjunto de estándares de desarrollo de software de la IEEE también contempla el rubro de desarrollo de sitios web, por lo que debido a lo anterior; se presenta como trabajo futuro la implementación de estándares de calidad para desarrollo de software de Internet propuestos por la IEEE.

BIBLIOGRAFÍA

- [IEEE, 2004] IEEE. "Guide to the Software Engineering Body of Knowledge", 2004.
- [Moore, 1999]. Moore, James W."Una colección integrada de estándares de calidad". Publicación de la IEEE . Noviembre-diciembre1999
<http://www.asq509.org/ht/action/GetDocumentAction/id/490>
- [Reyes, 2005]. A. Percy Reyes Paredes. Construyendo software de Calidad. 13/Jul/2005 (07/07/2005)
URL:
(www.elguille.info/colabora/NET2005/Percynet_ConstruyendoSoftCalidad.htm)
- [e-Economía, 2004] e-Economía. Programa para el Desarrollo de la Industria del Software, versión 1.2. Secretaría de Economía.
<http://www.software.net.mx/sniiti/infoagent.aspx?docid=200>
- [Robles, 2003]. Robles, Gregorio "II Jornadas de Software Libre de Segovia". 15/abr/2003, URL:
(www.gsync.escet.urjc.es/~grex/Segovia200204.pdf)
- [Rea, 2003]. Rea Cortés Joel Iván "Ingeniería de Software Asistida por Computadora, Nov/2003, URL:
(www.mail.udlap.mx/~pgomez/ing_sw/case.pdf)
- [Fernández, 1995] Fernández Carrasco Oscar M., García León Delba, Beltrán Benavides Alfa. "Un Enfoque Actual Sobre la Calidad del Software". ACIMED 3(3) Sep-dic 1995.
- [Moore, 2006]. Moore Jim, "ISO 12207 and Related Software Life-Cycle Standards", 2006, URL:
<http://www.acm.org/tsc/lifecycle.html>
- [Beauchemin,1998]. Dr. S.S. Beauchemin. "T10E IEEE SPMP UNIT 1998". 25/08/98, URL (<http://www.scs.carleton.ca/~beau/PM/T10E-SPMP.html>)
- [Bowne, 2005]. Bowne, Gregory M ."Software Standard's ", URL:
(<http://members.aol.com/kaizensepg/standard.htm>)
- [García, 2005] García Barranco, Irma María. "Modelos de Procesos para la Pyme". Revista Software Gurú. Edición Nov-Dic. 2005. Pág. 32

- [Bruegge, 2002] Bruegge Bernd, Dutoit Allen H. "Ingeniería de Software Orientado a Objetos". Ed. Prentice Hall. Págs.
- [Jenkins, 2003] Jenkins Marcelo. "Enseñando Aseguramiento de la Calidad del Software en un Programa de Posgrado". XI Congreso Iberoamericano de Educación Superior en Computación en el marco de CLEI 2003.
<http://ism.dei.uc.pt/ribie/pt/textos/doc.asp?txtid=22>
- [Fernández, 1999] Oscar M. Fernández Carrasco¹, Delba García León² y Alfa Beltrán Benavides³, URL:
(http://www.bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm)
- [Cueva, 1999] Cueva Lovelle Juan Manuel Calidad del Software. Conferencia, 21 de Octubre de 1999 Grupo GIDIS Universidad Nacional de la Pampa,
URL:(http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF)
- [Fernández, 2001] Luis Fernández, Grupo de Trabajo de Madrid; "Calidad del Software", May 23, 2001, URL: (www.ati.es/gt/calidad-software/presentacion.htm.)
- [Barrios, 2006], Manuel Barrios Solórzano, "Ingeniería de Software Distribuido", 2006, URL:
(<http://www.infor.uva.es/%7Embarrio/ISO1/Teoria/ISO1Introduccion.pdf>)
- [Treviño,] Treviño, Sandra P., El uso de las SSD en las MiPyMes Mexicanas. 2005, URL:
(<http://www.gestiopolis.com/canales2/gerencia/1/ssdsandra.htm>)
- [Sexto, 2006] Luis Felipe Sexto, "El Ciclo de la Calidad", 2006, URL:
(<http://www.estrucplan.com.ar/articulos/verarticulo.asp?IDArticulo=755>)

PARTE IV

ANEXOS

DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS

ISC Claudia Rodríguez Lemus

Contraloría Municipal de Celaya, Gto.

Agosto 2004

Contenido:

<u>1. Introducción</u>	3
<u>1.1 Propósito del Sistema</u>	3
<u>1.2 Alcance del sistema</u>	3
<u>1.3 Objetivos y criterios de éxito del proyecto</u>	4
<u>1.4 Definiciones, siglas y abreviaturas</u>	5
<u>1.5 Referencias</u>	5
<u>2. Sistema Actual</u>	6
<u>2.1 Actores</u>	6
<u>2.2 Tipos de Declaraciones</u>	6
<u>2.3 Proceso de entrega de Formatos</u>	7
<u>2.4 Proceso de Llenado de Formatos</u>	7
<u>2.5 Proceso de Recepción de Formatos</u>	8
<u>2.6 Proceso de cálculo de Balanza</u>	8
<u>2.7 Proceso de Consulta de Servidores Públicos Morosos</u>	9
<u>2.8 Proceso de Almacenamiento de Información</u>	9
<u>3. Sistema Propuesto</u>	10
<u>3.1 Panorama</u>	10
<u>3.2 Requerimientos Funcionales</u>	12
<u>3.3 Requerimientos no funcionales</u>	18
<u>3.3.1 Interfaz de Usuario y Factores Humanos</u>	18
<u>3.3.2 Documentación</u>	22
<u>3.3.3 Consideraciones de hardware</u>	22

<u>3.3.4 Características de desempeño</u>	23
<u>3.3.5 Manejo de errores y condiciones extremas</u>	23
<u>3.3.6 Cuestiones de calidad</u>	23
<u>3.3.7 Modificaciones al sistema</u>	23
<u>3.3.8 Ambiente físico</u>	24
<u>3.3.9 Cuestiones de seguridad</u>	24
<u>3.3.10 Cuestiones de recursos</u>	24
<u>4. Seudorrequerimientos</u>	25
<u>5. Modelos del Sistema</u>	26
<u>5.1 Escenarios y Casos de Uso.</u>	26
<u>5.2 Modelo de objetos</u>	29
<u>5.2.1 Diccionario de Datos</u>	29
<u>5.2.2 Diccionario de Clases</u>	31
<u>6. Glosario</u>	35

1. Introducción

1.1 Propósito del Sistema

La dependencia municipal de la Contraloría solicitó un sistema informático para la Declaración de Servidores Públicos (SIDS); el se desarrollará para almacenar y controlar la información de los bienes de los servidores públicos del municipio de Celaya, Gto. y de sus dependientes.

1.2 Alcance del Sistema

El alcance del SISP será el siguiente:

- i. Contará con un catálogo de Servidores públicos que estén activos
- ii. A su vez contará con un catálogo de Servidores Inactivos
- iii. Verificará el tipo de declaración correspondiente al servidor público:
 - iii.1 Se deberá realizar al menos una declaración inicial y una final
 - iii.2 Se podrán hacer hasta 3 declaraciones anuales en caso de que el servidor público complete los 3 años en su cargo.
 - iii.3. En caso de que el servidor público cambie de cargo en ese mismo período, el SIDS verificará que primero cierre la declaración de su cargo anterior para comenzar nuevamente las declaraciones con el nuevo cargo.

- iv. Solicitará la información para las declaraciones de los dependientes del servidor público.
- v. El SIDS calculará e imprimirá una balanza con la declaración actual, la cual será almacenada para futuras referencias.
- vi. SIDS producirá una lista con los servidores públicos que faltasen de entregar su declaración

1.3 Objetivos y criterios de éxito del proyecto

1.3.1 Objetivo General:

Construir un documento de Especificación de requerimientos para el sistema SIDS a desarrollar para la Contraloría municipal de Celaya, Gto.

1.3.2 Objetivos Particulares:

1.3.2.1 Determinar a los actores

1.3.2.2 Realizar entrevistas para encontrar hechos

1.3.2.3 Realizar observaciones y revisión de documentos

1.3.2.4. Clasificar requerimientos

1.3.2.5 Entregar documento de especificación.

1.3.3 Criterios de Éxito del Proyecto.

1.3.3.1 El SIDS contará con un catálogo de la información personal de cada servidor público y de cada uno de sus

dependientes, por lo que solo insertará esa información una sola vez, y para futuras declaraciones, solo hará una búsqueda a la base de datos.

1.3.3.2 El SIDS permitirá realizar modificaciones en cada declaración de algún dato personal del servidor público o de sus dependientes

1.3.3.3 Su entorno será amigable, por lo que el servidor público podrá hacer su declaración sin que esté presente Contraloría

1.3.3.4 Presentará una balanza con el concentrado de la información de cada declaración.

1.3.3.5 Llevará un control en las declaraciones

1.4 Definiciones, siglas y abreviaturas

- SIDS: Sistema informático para Declaración Patrimonial.
- Servidor Público: Persona que trabaja para el municipio.
- Declarante: Servidor Público con obligaciones para declarar sus bienes patrimoniales

1.5 Referencias

1. Bruegge. Ingeniería de Software Orientado a Objetos. Prentice Hall. Año 2002.
2. Sommerville Ian. Ingeniería de Requerimientos.
3. Documento de la IEEE para el estándar 830
4. Documento ISO/IEC 12207

2. Sistema Actual

2.1 Actores

Los actores que se encontraron para este sistema se describen a continuación.

- **Contraloría:** Los empleados del departamento de contraloría que llevan a cabo la recepción y control de las declaraciones de los servidores públicos
- **Servidores Públicos:** Los servidores que trabajan en el municipio y están obligados a entregar sus declaraciones patrimoniales a Contraloría
- **Dependientes:** Pueden ser tanto el cónyuge como sus dependientes del servidor público con bienes personales que requieran declarar.
- **Desarrollador:** Ingeniero de Software que se hará cargo de la construcción del SIDS
- **SIDS:** Sistema informático para Declaración de Servidores

2.2 Tipos de Declaraciones

Las declaraciones pueden ser de 3 tipos: Declaración Inicial, Declaración Anual y Declaración Final.

- La Declaración Inicial se puede llevar a cabo en dos situaciones: Cuando el Servidor Público comienza un nuevo cargo o cuando se comienza una nueva administración.
- La Declaración Anual se lleva a cabo cuando ha pasado un año después de su última declaración.
- La Declaración Final se lleva a cabo cuando el servidor público ha terminado su cargo ya sea por fin de administración o por ser cambiado de puesto o por ser dado de baja.

Restricciones:

- Solo se realiza una declaración inicial en cada administración o en cada nuevo cargo.
- Existen ocasiones en donde el servidor público es removido de su cargo antes de terminar el año, ya sea porque fue dado de baja como empleado de gobierno o porque fue cambiado de puesto. En ese caso, no realizará una declaración anual, sino una final.
- Las declaraciones Anuales pueden ser máximo 3 por administración.
- Solo se realiza una declaración final y se puede llevar a cabo ya sea porque terminó la administración, porque el servidor público cambió de puesto o porque fue dado de baja.
- En cada nueva administración o nuevo cargo público, se comienza con una nueva declaración Inicial.



Contraloría Municipal
Celaya, Gto.

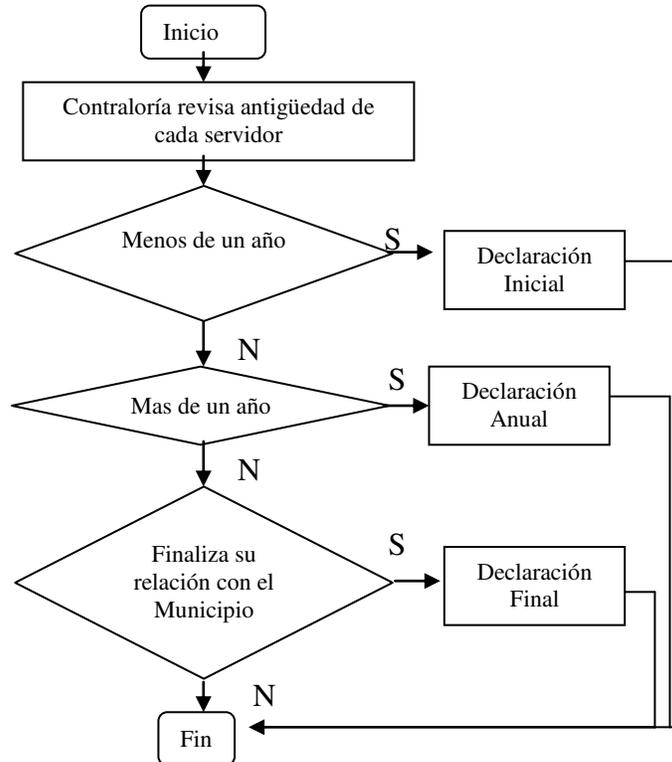
Nombre del proceso: Determinación de Tipo de Declaración

Código F.1.1.4

Finalidad:

El objetivo es que el departamento de Contraloría determine que el Tipo de Declaración a realizar

Logigrama



Entradas:

Antigüedad

Salidas:

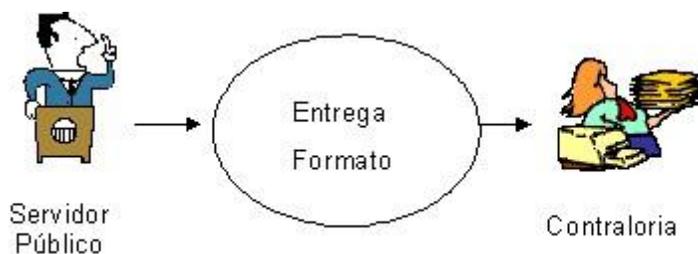
Tipo de declaración.

Actividades y Resultados:

- Los procesos inician cuando Contraloría revisa antigüedad
- Los tipos de declaración son seleccionados

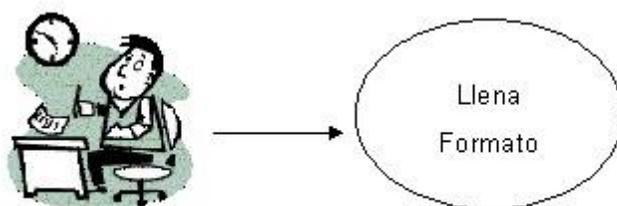
2.3 Proceso de Entrega de Formatos

El departamento de Contraloría entrega el formato correspondiente a cada Servidor público, ya sea el de la declaración inicial, anual o final; para su llenado:



2.4 Proceso de llenado de Formatos

El servidor público debe llenar los formatos de acuerdo a los ingresos y egresos que tuvo en el año, en caso de ser una declaración anual. En caso de ser una declaración inicial, realiza una declaración de sus bienes patrimoniales, de sus deudas y de sus inversiones.



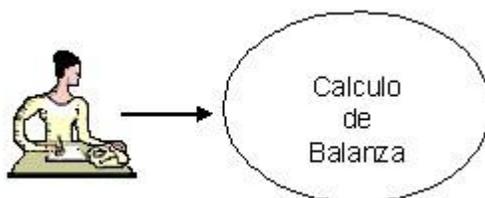
2.5 Proceso de Recepción de Formatos

Los servidores públicos tienen hasta el último día hábil del mes de mayo para entregar su formato de declaración patrimonial al departamento de contraloría



2.6 Proceso de cálculo de Balanza

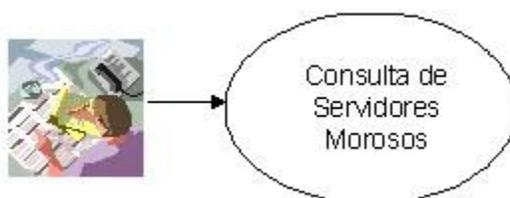
El departamento de contraloría se dispone a hacer el cálculo de la balanza de la declaración que acaba de recibir. Esta se realiza sumando todos los ingresos junto con los valores patrimoniales existentes y restando a este resultado los egresos.



En caso de que se encuentren anomalías, se avisa inmediatamente al servidor público para que las corrija.

2.7 Proceso de Consulta de Servidores Públicos Morosos

Después de realizar el cálculo de la balanza a todas las declaraciones recibidas en ese período, se procede a hacer una lista de todos los servidores que no entregaron su declaración patrimonial. Posteriormente, se realiza un documento en donde se invita al servidor público a que entregue su declaración. En caso de hacer caso omiso a esta llamada de atención, el jefe del departamento de contraloría determinará el proceso a seguir con respecto a este servidor.



2.8 Proceso de Almacenamiento de Información

Después de haber calculado la balanza de la declaración y verificado los datos, se procede a guardar los formatos en su carpeta correspondiente del servidor público para futuras referencias.



3. Sistema Propuesto

3.1 Panorama

Después de analizar el sistema actual, se encontraron los siguientes problemas:

- Debido a que la entrega de las declaraciones es de forma manual, en una declaración correcta (libre de errores por parte del servidor público) una persona del departamento de contraloría que lleve a cabo el cálculo de la balanza y almacenamiento de la declaración para un servidor público tarda un día. Se asignan 3 personas dentro del departamento de contraloría para este proceso. Actualmente están obligados a entregar su declaración patrimonial 120 servidores públicos, por lo que el número total de días hábiles para este proceso, son 40 días como mínimo.
- Los servidores públicos cometen muchos errores al momento de llenar sus declaraciones y Contraloría se da cuenta hasta el momento en que se está calculando la balanza, por lo que se debe regresar nuevamente la declaración al servidor público para que la modifique. Lo anterior provoca que el se atrase más el proceso de cálculo de la balanza. Este retraso va a depender del tiempo que tarde el servidor público en entregar nuevamente su declaración.
- Contraloría tarda aprox. un día más en hacer una lista con todos los servidores públicos que no entregaron su declaración.
- El tiempo que tarda Contraloría en elaborar el documento de invitación a entregar la declaración, variará del número de servidores que no hayan declarado, sin embargo, el tiempo promedio está estimado en medio día.
- En promedio, una declaración de un servidor público que tuvo errores y que fueron corregidos, se almacena en Contraloría en el mes de agosto.
- La mayoría de los Servidores Públicos solicitan al departamento de Contraloría su declaración anterior para llenar la actual, por lo que se le deben prestar los formatos anteriores que después tardan en entregar o extravían.
- En cada declaración, el servidor público debe introducir sus datos personales completos, lo que resulta ser tedioso y alarga el tiempo de entrega.

Debido a los problemas anteriores, el equipo de desarrolladores propone al departamento de Contraloría la implementación del sistema SIDS el cual contará con las siguientes características:

- Interfaces amigables y con controles que evitarán cometer errores a los Servidores Públicos lo que agilizará el trámite.

- Cálculo de la balanza que podrá ser realizada por el mismo servidor público en el momento en que termine de introducir toda su información.
- Concentrado de las declaraciones anteriores del Servidor Público para referencias
- El departamento de Contraloría solo verificará los datos de la declaración entregada
- Podrán imprimir la balanza de cada servidor público.
- El tiempo de almacenamiento de la información correcta será de 3 días hábiles como máximo.
- El tiempo de consulta e impresión de los servidores públicos morosos, será de 1 hora como máximo.
- Los datos personales del Servidor Público serán almacenados en la base de datos del sistema, con posibilidad de modificarse.

3.2 Requerimientos Funcionales

A continuación se presenta una tabla con los requerimientos funcionales encontrados:

#	Requerimiento	Necesidad	Estado	Versión
1	Captura de los datos personales del servidor público y de sus dependientes	E	Cerrado	4
2	Captura de adquisición de bienes muebles y de bienes inmuebles	E	Cerrado	5
3	Captura de la venta o donación de sus bienes muebles y de sus bienes inmuebles	E	Cerrado	5
4	Captura de las inversiones del Servidor Público	E	Cerrado	1
5	Declaración de pagos realizados	E	Cerrado	1
6	Captura de Dependencias	C	Cerrado	1
7	Captura de Departamentos	C	Cerrado	1
8	Captura del tipo de Declaración	E	Cerrado	3
9	Cálculo de la Balanza	E	Cerrado	5
10	Reporte de servidores públicos que no declararon	C	Abierto	2

Donde

Necesidad:

- E, cuando el requerimiento es esencial o indispensable para la aceptación del sistema;
- C, cuando el requerimiento se considera condicional o bien deseable pero no indispensable para la aceptación del sistema
- O, si el requerimiento es opcional sugerido por el usuario.>

Estado :

- C: Cerrado, cuando se considera que el requerimiento se encuentra lo suficientemente claro y definido, como para pasar a la siguiente etapa del análisis.
- A: Abierto, cuando se considera que el requerimiento está pendiente de mayor definición.

Versión:

Indica el número de versión del requerimiento.

A continuación se describe a detalle cada requerimiento:

- Proceso para la captura de los datos personales del servidor público y de sus dependientes: En este proceso se solicitarán los siguientes datos por persona:

3.2.1 Nombre y apellidos

3.2.2 RFC

3.2.3 Dirección: Calle, número, colonia y ciudad.

3.2.4 Departamento y Dependencia en la que labora

3.2.5 Edad

3.2.6 Sueldo mensual

3.2.7 Prestaciones

3.2.8 Honorarios

- Proceso para la captura de adquisición de bienes muebles y de bienes inmuebles.

Las adquisiciones se podrán registrar como compra, donación, herencia, Otros. Especificar

En este proceso se solicitarán los siguientes datos:

1. Para Bienes muebles

Se le considerará bien mueble a: Joyas y Obras de Arte, Muebles del Hogar, Semovientes (Cabezas de Ganado), y Vehículos.

1.1 Para todos los casos, se solicitará su valor económico.

1.2 En caso de vehículos, se solicitará además:

Marca, Modelo, placas

2 Para Bienes inmuebles

Se le considerará Bien Inmueble a: Casa Habitación, Departamento, Lote, Local Comercial, Otros. Especificar. Como datos se solicitarán:

Número de Cuenta del predial, Valor comercial, tipo de inmueble, superficie en metros cuadrados, construcción en metros cuadrados, Clase de operación (Compra, donación, herencia, otra. Especificar), fecha de operación y dirección del bien inmueble.

- Proceso para la captura de la venta o donación de sus bienes muebles y de sus bienes inmuebles

Se solicitará el tipo de operación que se realizó: Venta o donación.

Posteriormente:

1. Para bienes muebles el SIDS solicitará:

1.1 El nombre del bien mueble

1.2 En caso de vehículo, el número de placas

2 Para bienes inmuebles el SIDS solicitará:

2.1 El número de Predial.

3 El SIDS con los datos anteriores buscará los datos en su base de datos y en caso de encontrarlos, los mostrará en la forma, en caso contrario, se enviará una forma con mensaje de error.

- Proceso para la captura de las inversiones del Servidor Público.

1 El SIDS solicitará el registro de Apertura o Variación en su caso.

2 Solicitará el tipo de inversión: Cheques, Ahorro, Valores, Tarjetas de Inversión, Membresías., Seguros, Acciones, Partes Sociales, Capital Invertido, Otros.

3 Solicitará el número de la cuenta. Nombre de la institución en donde se encuentre su capital, Saldo o Monto Actual, Cantidad incrementada o disminuida en su caso.

- Proceso para la declaración de pagos realizados

El SIDS solicitará el tipo de gravamen: Créditos Hipotecarios, Préstamos, Compras a Crédito, Tarjetas de Crédito, Otros., Fecha de operación, el plazo (en meses), pagos netos, institución acreedora, número de tarjeta y saldo del adeudo.

- Proceso para la captura de Dependencias

Proceso para uso exclusivo de Contraloría. El SIDS solicitará la captura de todas las dependencias que contenga el gobierno municipal y las almacenará en un catálogo.

- Proceso para la captura de Departamentos

Proceso para uso exclusivo de Contraloría. El SIDS solicitará el nombre de la dependencia que contiene a ese departamento y la buscará en su base de datos, en caso de encontrarla, Contraloría podrá insertar los departamentos de la Dependencia, en caso contrario, el SIDS mostrará un mensaje de error y no permitirá la captura de los Departamentos.

- Proceso para la captura del tipo de Declaración

El SIDS solicitará el tipo de declaración que se desee realizar y verificará en su base de datos. En caso de que si corresponda, el sistema permitirá continuar con la declaración y registrará el tipo declaración, en caso contrario, el SIDS enviará un mensaje de error y no permitirá la continuación de la declaración.

- Proceso para el Cálculo de la Balanza

El sistema solicitará el RFC del declarante y el tipo de declaración a calcular. En caso de que ya exista esa declaración el sistema

buscará las adquisiciones y ventas o donaciones de sus bienes muebles y de sus bienes inmuebles, de sus pagos y de sus variaciones en sus inversiones. Posteriormente, calculará la nueva balanza y permitirá salvarla y/o imprimirla.

- Proceso para reportar a los servidores públicos que no declararon

1. Reporte con la lista de servidores

El SIDS solicitará el período que se desee verificar y buscará a los servidores que no tengan registrado su declaración en ese período. Posteriormente, los mostrará en la forma y permitirá imprimirlos.

2. Documento para solicitar la declaración por servidor

El SIDS contará con un formato ya establecido y solo anexará los datos del servidor público que no haya efectuado su declaración, y lo imprimirá para ser enviado. A continuación se muestra el formato:

Guanajuato, Gto.,
Oficio No:
Sección: Dirección General de Asuntos Jurídicos.
Asunto: Requerimiento de declaración patrimonial

Ciudadano

Ex-
Presidencia Municipal

Presente

La declaración de situación patrimonial, es un elemento importante para brindar a la ciudadanía confianza respecto a la actuación honesta y transparente de quienes desempeñamos algún cargo en la Administración Pública. Por tal motivo, la Ley de Responsabilidades de los Servidores Públicos del Estado establece como una obligación presentarla ante esta Secretaría.

De la revisión de nuestros archivos se observó que no ha entregado usted su declaración de situación patrimonial por lo que se le requiere para que esta omisión sea corregida antes de 3 días contados a partir de la Fecha en que se reciba el presente, y así dar cumplimiento a lo establecido en los artículos 34, 35, y 36 de la Ley de Responsabilidades de los Servidores Públicos del Estado.

Sin otro particular por el momento, reitero a usted las seguridades de mi consideración distinguida.

A t e n t a m e n t e
Guanajuato, Tierra de Oportunidades

LIC.
El Director de Asuntos Jurídicos

Con Copia

-

Conocimiento y efectos del párrafo segundo del artículo 36 de la Ley de Responsabilidades de los Servidores Públicos del Estado Presente.

- C. Lic. Antonio Morales Moreno.- Director de Asesoría Legal y Situación Patrimonial.- Presente

- Archivo.

3.3 Requerimientos no funcionales

3.3.1 Interfaz de Usuario y Factores Humanos

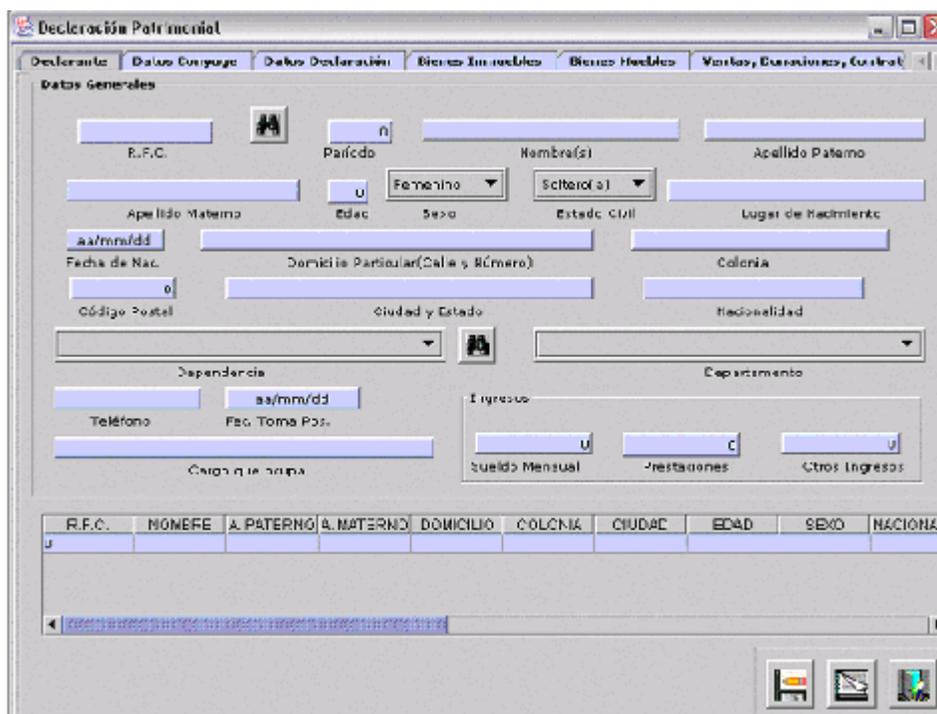
Para evitar problemas de ubicación, el diseño de las formas se hará en cejillas en donde el servidor y el departamento de contraloría podrán insertar información o consultarla. A continuación se muestra la pantalla para la declaración patrimonial del servidor público en el formato de especificación de pantallas.

Pantalla “Declarante”.

Objetivo

Esta pantalla se utilizará para la consulta, inserción ó actualización de los datos del declarante (servidor público) y mostrará el listado de los declarantes que estén dados de alta en el sistema.

Imagen



Descripción

Campo	Tipo	Comentario
RFC	Alfanumérico	Campo necesario para identificar los registros.
Periodo	Numérico	Campo necesario para conocer número de período de la auditoría
Nombre	Alfanumérico	Campo necesario para el nombre del declarante
aPaterno	Alfanumérico	Campo necesario para el apellido paterno del declarante
aMaterno	Alfanumérico	Campo necesario para el apellido paterno del declarante
edad	Numérico	Campo utilizado para la edad del declarante.
INacimiento	Alfanumérico	Campo necesario para especificar el lugar de nacimiento del declarante
fNacimiento	Fecha	Campo necesario para especificar la fecha de nacimiento del declarante
domicilio	Alfanumérico	Campo necesario para especificar la calle y número del domicilio del declarante
Ciudad	Alfanumérico	Campo necesario para especificar la ciudad y estado del declarante
Nacionalidad	Alfanumérico	Campo necesario para especificar la nacionalidad del declarante

Dependencia	Alfanumérico	Campo necesario para identificar la dependencia a la que pertenece el declarante
departamento	Alfanumérico	Campo necesario para identificar el departamento al que pertenece el declarante
Teléfono	Alfanumérico	Campo necesario para especificar el número de teléfono del declarante
tPosecion	Fecha	Campo necesario para especificar la fecha de toma de posesión del declarante
Cargo	Alfanumérico	Campo necesario para especificar el cargo que ocupa el declarante
Sueldo	Numérico	Campo necesario para especificar el sueldo del declarante
Prestaciones	Numérico	Campo necesario para especificar el ingreso por prestaciones del declarante
otros	Numérico	Campo necesario para especificar el ingreso por otro tipo del declarante

Control	Tipo	Comentario
bBuscaRFC	Button	Objeto que busca los datos del declarante
Sexo	Combobox	Lista que muestra y permite elegir el sexo del declarante
eCivil	Combobox	Lista que muestra y permite elegir el estado civil del declarante
cDependencia	Combobox	Lista que muestra y permite elegir la dependencia a la que perteneces el declarante
bBuscaDep	Button	Objeto que busca los departamentos que pertenecen a la dependencia elegida y los carga en el objeto cDepartamento
cDepartamento	Combobox	Lista que permite elegir el departamento al que corresponde el declarante
tDatos	Table	Tabla que muestra una lista de todos los campos del declarante. Estos son cargados en el evento abrir pantalla
bActualizar	Button	Objeto que realiza inserción ó actualización de un registro
bLimpiar	Button	Objeto que limpia los campos de la pantalla
bCerrar	Button	Objeto que cerrar pantalla

Descripción de funcionalidad

Esta pantalla puede manipular la Tabla de la Base de Datos que contiene los datos del declarante, realizando las siguientes operaciones:

- Navegar entre todos los registros.
- Buscar un registro.
- Agregar un nuevo registro.
- Modificar el registro actual.

Bitácora de Cambios

Fecha	Versión	Descripción del Cambio	Autor
24/03/05	Ver. 1		Claudia Rodríguez Lemus

El proceso de alta de dependencias estará a cargo del departamento de Contraloría:



Municipio de Celaya, Gto.

DeclarPat 1.0
Especificación de Pantalla

Contraloría Municipal

Pantalla “Catálogo de Dependencias”.

Objetivo

Esta pantalla se utilizará para insertar o actualizar dependencias en el sistema

Imagen



Descripción

Campo	Tipo	Comentario
tDependencia	Alfanumérico	Campo necesario para insertar o actualizar dependencias

Control	Tipo	Comentario
bBusca	Button	Objeto que busca un registro
bActualiza	Button	Objeto que inserta ó actualiza un registro
bLimpia	Button	Objeto que limpia el campo tDependencia
bCierra	Button	Objeto que cierra la pantalla

Descripción de funcionalidad

Esta pantalla puede manipular la Tabla de la Base de Datos que contiene los datos de las dependencias, realizando las siguientes operaciones:

- Navegar entre todos los registros.
- Buscar un registro.
- Agregar un nuevo registro.
- Modificar el registro actual.

Bitácora de Cambios

Fecha	Versión	Descripción del Cambio	Autor
25/03/05	Ver. 1		Claudia Rodríguez Lemus

Un departamento no podrá ser dado de alta si antes no se registro su dependencia:



Municipio de Celaya, Gto.

DeclarPat 1.0
Especificación de Pantalla

Contraloría Municipal

Pantalla “Catálogo de Dependencias”.

Objetivo

Esta pantalla se utilizará para insertar o actualizar departamentos en el sistema

Imagen



Descripción

Campo	Tipo	Comentario
tDepto	Alfanumérico	Campo necesario para insertar o actualizar departamentos

Control	Tipo	Comentario
bBusca	Button	Objeto que busca un registro
bActualiza	Button	Objeto que inserta ó actualiza un registro
bLimpia	Button	Objeto que limpia el campo tDepto
bCierra	Button	Objeto que cierra la pantalla

Descripción de funcionalidad

Esta pantalla puede manipular la Tabla de la Base de Datos que contiene los datos de los Departamentos, realizando las siguientes operaciones:

- Navegar entre todos los registros.
- Buscar un registro.
- Agregar un nuevo registro.
- Modificar el registro actual.

Bitácora de Cambios

Fecha	Versión	Descripción del Cambio	Autor
25/03/05	Ver. 1		Claudia Rodríguez Lemus

La balanza general mostrará todos los egresos e ingresos de la declaración por servidor público y permitirá imprimirla. Una restricción que tendrá es que no el usuario no podrá realizar modificaciones de algún dato, solo es de consulta para evitar incongruencias de la información.



Pantalla “Balanza General”.

Objetivo

Esta pantalla se utilizará para calcular la balanza de la declaración del servidor público. En ella se muestran el total de ingresos y egresos tanto del declarante como del cónyuge y al final calcula el total de la aplicación.

Imagen

Concepto	Declarante	Cónyuge y/o Dependiente	Total
1. Sueldos, honorarios y otras prestaciones (Bono, aguinaldo, compensaciones)	0	0	0
1B. Honorarios	0	0	0
2. Rentas, regalías, intereses y dividendos	0	0	0
3. Donaciones, herencias, intereses o legados	0	0	0
4. Préstamos hipotecarios, personales u otros	0	0	0
5. Venta de bienes inmuebles	0	0	0
6. Venta de bienes muebles y/o semovientes	0	0	0
7. Retiro de ahorro, valores, acciones, etc	0	0	0
TOTAL DE INGRESOS	0	0	0
1. Pago de adeudos	0	0	0
2. Adquisición de bienes inmuebles	0	0	0
3. Adquisición de muebles y/o semovientes	0	0	0
4. Saldo en bancos, valores, acciones u otros	0	0	0
5. Gastos generales	0	0	0
TOTAL DE APLICACIÓN	0	0	0

Descripción

Campo	Tipo	Comentario
Rfc	Alfanumérico	Campo necesario para buscar la declaración del servidor público
Fecha	Fecha	Campo necesario para registrar la fecha de la declaración
Período	Numérico	Campo necesario para registrar el número del período
tSueldoD y tSueldoC	Numérico	Campos que muestra los sueldos tanto del servidor como del cónyuge
tTotalSueldo	Numérico	Campo que muestra el cálculo del total de la suma de los sueldos
tHonorarioD y tHonorarioC	Numérico	Campos que muestra los honorarios tanto del servidor como del cónyuge
tRentaD y tRentaC	Numérico	Campos que muestra los ingresos por renta tanto del servidor como del cónyuge
tDonacionD y tDonacionC	Numérico	Campos que muestra los ingresos por donaciones tanto del servidor como del cónyuge
tPrestamoD y tPrestamoC	Numérico	Campos que muestra los ingresos por préstamos tanto del servidor como del cónyuge
tVMueblesD y tVMueblesC	Numérico	Campos que muestra los ingresos por venta de bienes muebles tanto del servidor como del cónyuge
tVInmueblesD y tVInmueblesC	Numérico	Campos que muestra los ingresos por venta de bienes inmuebles tanto del servidor como del cónyuge
tRetiroD y tRetiroC	Numérico	Campos que muestra los ingresos por algún retiro tanto del servidor como del cónyuge
tTotalHonor	Numérico	Campo que muestra el cálculo del total de la suma de los honorarios
tTotalRenta	Numérico	Campo que muestra el cálculo del total de la suma de las rentas
tTotalDonacion	Numérico	Campo que muestra el cálculo del total de la suma de las donaciones
tTotalPrestamo	Numérico	Campo que muestra el cálculo del total de la suma de los préstamos
tTotalVMueble	Numérico	Campo que muestra el cálculo del total de la suma de la venta de muebles
tTotalVInmueble	Numérico	Campo que muestra el cálculo del total de la suma de la venta de inmuebles
tTotalRetiro	Numérico	Campo que muestra el cálculo del total de la suma de los retiros
tTotalIngresosD y tTotalIngresosC	Numérico	Campos que muestran el total de los ingresos tanto del servidor como del cónyuge
tPagoD y tPagoC	Numérico	Campos que muestra los egresos por pago de adeudos tanto del servidor como del cónyuge
tAMuebleD y tAMuebleC	Numérico	Campos que muestra los egresos por compra de bienes muebles tanto del servidor como del cónyuge
tAlnmuebleD y tAlnmuebleC	Numérico	Campos que muestra los egresos por compra de bienes inmuebles tanto del servidor como del cónyuge
tSaldoD y tSaldoC	Numérico	Campos que muestra los egresos por pagos a bancos tanto del servidor como del cónyuge
tGastoD y tGastoC	Numérico	Campos que muestra los egresos por otros gastos tanto del servidor como del cónyuge
tTotalPago	Numérico	Campo que muestra el cálculo del total de la suma de los pagos
tTotalAMueble	Numérico	Campo que muestra el cálculo del total de la suma de compras de muebles
tTotalAlnmueble	Numérico	Campo que muestra el cálculo del total de la suma de compras inmuebles
tTotalSaldo	Numérico	Campo que muestra el cálculo del total de la suma de pagos a bancos

tTotalGasto	Numérico	Campo que muestra el cálculo del total de la suma de otros gastos
tTotalAplicacion	Numérico	Campo que muestra el cálculo de los totales de la suma tanto de los ingresos como de los egresos del declarante y del cónyuge.

Control	Tipo	Comentario
cTipo	ComboBox	Objeto que lista los diferentes tipos de declaración
bBusca	Button	Objeto que busca un registro
bActualiza	Button	Objeto que inserta ó actualiza un registro
bLimpia	Button	Objeto que limpia los campos de la pantalla
bCierra	Button	Objeto que cierra la pantalla

Descripción de funcionalidad

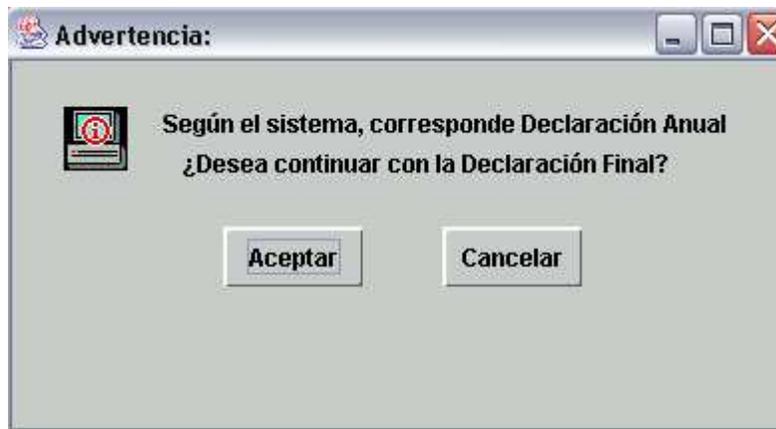
Esta pantalla calcula la balanza de la declaración y la almacena en la Tabla de la Base de Datos que contiene las balanzas, realizando las siguientes operaciones:

Buscar un registro que contenga la declaración
 Calcula balanza
 Imprime Balanza

Bitácora de Cambios

Fecha	Versión	Descripción del Cambio	Autor
25/03/05	Ver. 1		Claudia Rodríguez Lemus

Para mensajes de errores o advertencias, se presentará la siguiente forma.



En este ejemplo, el sistema ha encontrado que le corresponde al Servidor realizar su declaración anual, y pregunta al usuario si debe realizar su declaración final.

Para los servidores públicos morosos, el sistema solicitará el período a consultar y muestra una lista con los datos del servidor como son: RFC, nombre, apellido paterno, apellido materno, dependencia y departamento en el que labora

R.F.C.	NOMBRE	A. PATERNO	A. MATERNO	DEPENDEN...

3.3.2 Documentación

3.3.2.1 Entrega de sistema y solicitud de declaración.

El servidor público ya no requerirá de formatos en papel para llenar su declaración, por lo que solo se le entregará un documento en donde se especificará por parte de Contraloría que se le entregó el sistema y que se solicita su llenado de la declaración. El servidor público a su vez, firmará de recibido.

3.3.2.2 Recepción de declaración.

El servidor público regresará la base de datos de su declaración al departamento de Contraloría y este hará un documento de recibido con fecha y datos del servidor público.

3.3.2.3 Se contará con las balanzas impresas

3.3.2.4. Se contará con la lista impresa de los servidores morosos.

3.3.2.5 Se realizará un manual de usuario para servidores públicos y un manual de usuario para el departamento de Contraloría.

3.3.3 Consideraciones de hardware

Los requerimientos mínimos para equipo de cómputo del servidor público son:

1. Procesador pentium o athlon a 533 Mhz.
2. Espacio en Disco Duro de 100Mb (dependerá de los registros que se introduzcan)
3. Memoria RAM 128MB
4. Lector de CD ROM
5. Floppy 3.5"

Los requerimientos mínimos para el departamento de Contraloría son los mismos que para el Servidor, más las consideraciones de que el sistema trabajará en red, por lo que se requiere además:

1. La creación de 1 usuario con privilegios de administrador
2. La creación de 3 usuarios con privilegios para introducir los datos.
3. Una computadora configurada como servidor que contendrá la base de datos.
4. Al menos el servidor conectado a un UPS
5. Un switch de 4 puertos para hacer la conexión de la red.
6. Cable UTP nivel 5 para conectar las 3 computadoras con el switch.

3.3.4 Características de desempeño

Con las características mencionadas en el punto anterior, el sistema se ejecutará

3.3.5 Manejo de errores y condiciones extremas

El sistema validará las entradas para evitar la introducción incorrecta de los datos, y manejará procesos de respaldo y restauración para casos en donde se pierda la información.

3.3.6 Cuestiones de calidad

La determinación de requerimientos fue apoyada por el estándar de calidad de la IEEE 830. El ciclo de vida del sistema estará regido por el estándar de calidad ISO/IEC 12207, con lo que se asegura el correcto levantamiento de requerimientos y la solución adecuada para el cliente. El cableado de la red será instalado de acuerdo a los estándares de calidad de cableado estructurado.

3.3.7 Modificaciones al sistema

Se tiene contemplado para la siguiente versión subir el sistema a internet para que el servidor público pueda realizar su declaración en tiempo real.

3.3.8 Ambiente físico

El jefe del departamento contará con una caja de seguridad para guardar los respaldos de la información en su oficina. La entrada al área de la red será restringida.

3.3.9 Cuestiones de seguridad

En cuanto al departamento de Contraloría, el sistema permitirá crear usuarios con password para tener permisos para acceder el sistema. De estos usuarios, solo habrá uno con permiso de súper usuario para realizar cualquier cambio y será asignado al jefe del departamento.

En cuanto a los Servidores Públicos, solo se anexará una copia con la parte de sus datos del servidor público. Y el departamento de Contraloría tendrá la base de

datos original, por lo que antes de subir la información que el servidor envía, esta se debe verificar para proteger que no se hayan hecho cambios y altere o no coincida con la información que ya se tiene en la base de datos.

Además el sistema contará con un proceso de respaldo y recuperación en caso de siniestros.

3.3.10 Cuestiones de recursos

El municipio conoce de la importancia y necesidad de este sistema, por lo que ha destinado los recursos económicos necesarios para llevar a cabo el sistema.

4. Pseudorequerimientos

El diseño de la forma para la captura de las declaraciones se llevará a cabo uniendo todas las formas de captura en una sola, ya que así lo solicitó Contraloría. Se aclara que se considera pseudorequerimiento el diseño de la forma, no los datos contenidos en ella:

Declaración Patrimonial

Declarante Datos Conyuge Datos Declaración Bienes Inmuebles Bienes Muebles Ventas, Donaciones, Contrat

Datos Generales

R.F.C. Período Nombre(s) Apellido Paterno

Apellido Materno Edad Sexo Estado Civil Lugar de Nacimiento

Fecha de Nac. Domicilio Particular(Calle y Número) Colonia

Código Postal Ciudad y Estado Nacionalidad

Dependencia Departamento

Teléfono Fec. Toma Pos. Ingresos

Cargo que ocupa Sueldo Mensual Prestaciones Otros Ingresos

R.F.C.	NOMBRE	A. PATERNO	A. MATERNO	DOMICILIO	COLONIA	CIUDAD	EDAD	SEXO	NACIONAL
0									

5. Modelos del Sistema

5.1 Escenarios y Casos de Uso.

5.1.1 Actores:

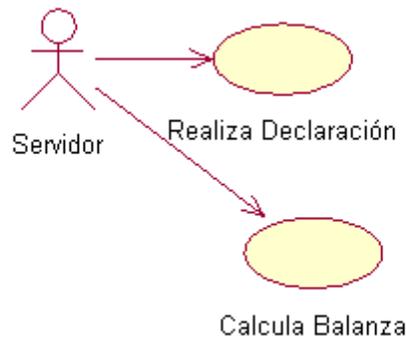


5.1.2 Escenarios y casos de uso:

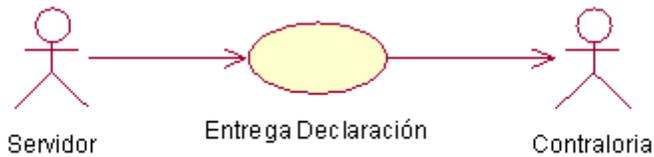
A continuación se presentan los Diagramas de Caso de uso del sistema Propuesto junto con sus escenarios:



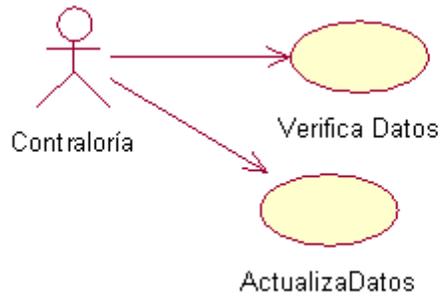
Primeramente el Departamento de Contraloría entrega el sistema con una base de datos limpia en caso de ser declaración inicial, de lo contrario, con los datos introducidos de una declaración anterior.



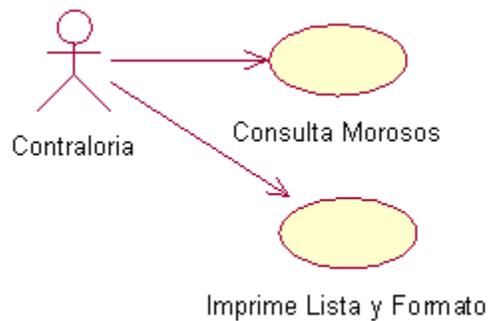
El servidor realiza su declaración en su computadora, calcula la balanza y guarda los cambios en un disquette



El servidor regresa el disco a Contraloría con la información de su declaración



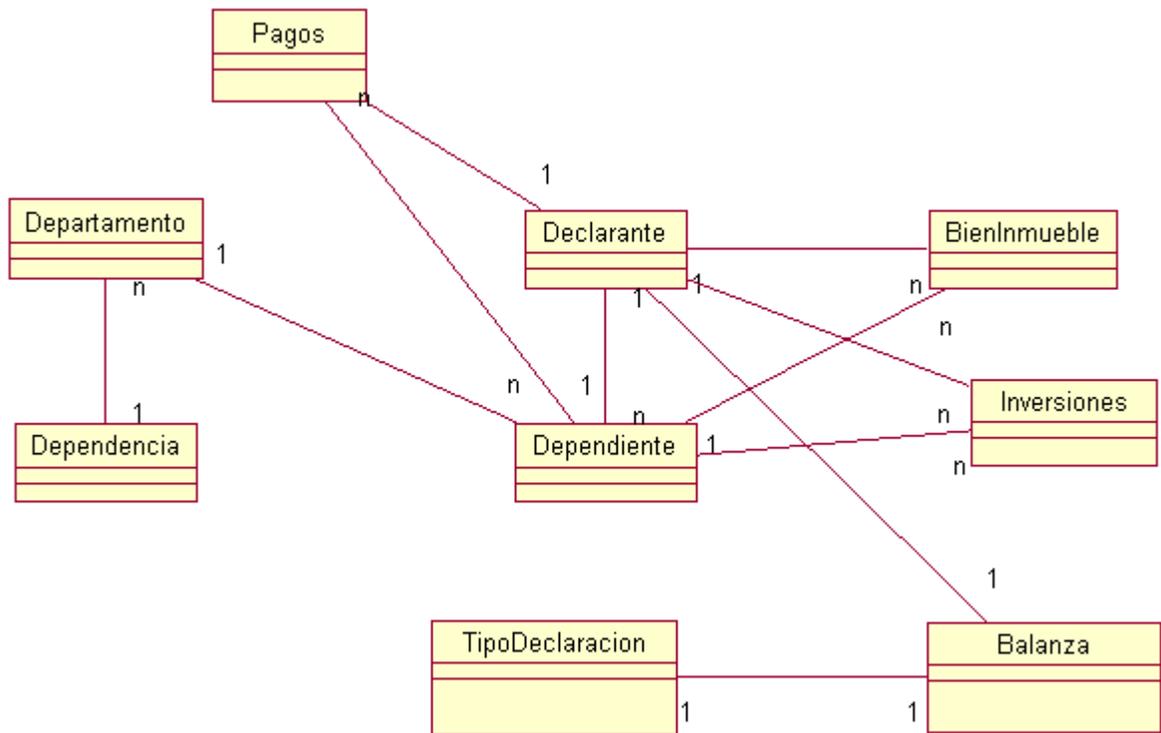
Contraloría verifica los datos y en caso de estar correctos, actualizará la base de datos, de lo contrario, verificará con el Servidor.



Contraloría hace una búsqueda de los servidores morosos e imprime un documento en donde invita al servidor a realizar su declaración

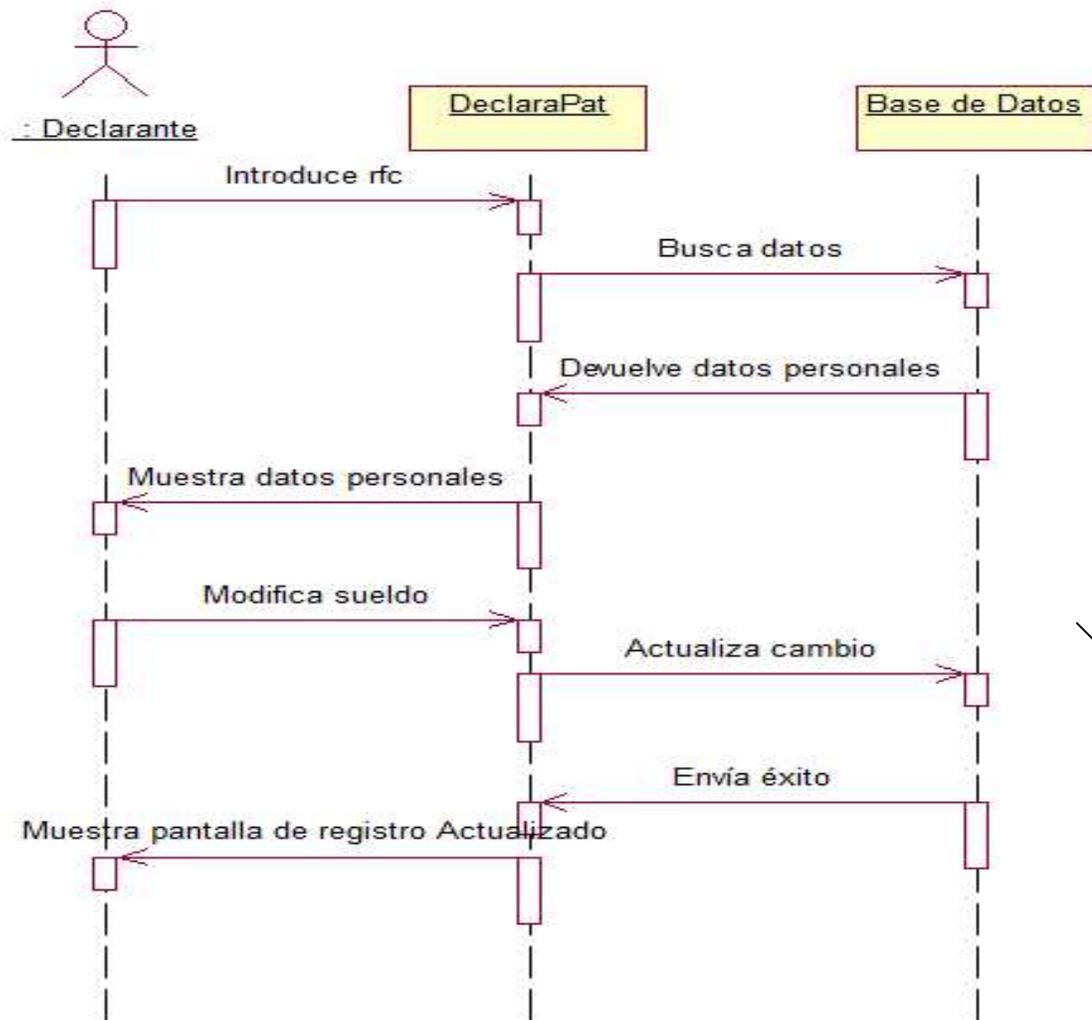
5.1.2 Diagrama de Clases

Se muestran las clases principales que utilizarán el sistema y sus relaciones



5.1.3 Diagrama de Secuencia

Se muestra la secuencia del proceso actualiza sueldo por el declarante.



5.3 Modelo de objetos

A continuación se presentará el diccionario de datos y el de clases:

5.3.1 Diccionario de Datos

A continuación se muestra el diccionario de datos

DICCIONARIO DE DATOS

TABLA							
Nombre :	Declarante		Módulo	Declaracion	Nombre PK :	refeca_pad	
Descripción:	Contiene los datos personales del declarante						
CAMPOS							
Nombre	Tipo y Longitud	PK	FK	CC	Not Null	Default	Descripción
REFECA_PAD	Texto	X			X		RFC del Declarante
NUMPER_PAD	Texto				X		Numero de Periodo a declarar
NUMFOL_PAD	Numerico				X		Numero de Folio
APEPAT_PAD	Texto				X		Apellido Paterno Declarante
APEMAT_PAD	Texto				X		Apellido Materno Declarante
NOMBRE_PAD	Texto				X		Nombre del Declarante
CVECAR_PAD	Texto		X		X		Clave de cargo
DEPEN_PAD	Texto		X		X		Dependencia
EDOCIV_PAD	Texto				X		Estado civil
SEXOMF_PAD	Texto				X		Sexo
DOMICI_PAD	Texto				X		Direccion Calle
COLONI_PAD	Texto				X		Colonia
CODPOS_PAD	Texto				X		Codigo Postal
TELEFO_PAD	Texto				X		Telefono
CIUDAD_PAD	Texto				X		Ciudad
LUGNAC_PAD	Texto				X		Lugar de Nacimiento
EDADTI_PAD	Numerico				X		Edad
FECNAC_PAD	Fecha				X		Fecha de nacimiento

FECPOS_PAD	Fecha				X		Fecha de toma de Posesión
FECBAJ_PAD	Fecha				X		Fecha de Baja
DEPTO_PAD	Texto		X		X		Departamento
SUECON_PAD	Numerico				X		Sueldo
NACION_PAD	Texto				X		Nacionalidad
OFIPRO_PAD	Texto				X		Otros Ingresos
HONOR_PAD	Numerico				X		Honorarios
PRESTA_PAD	Numerico				X		Prestaciones

TABLA							
Nombre :	Muebles		Módulo	Declaracion	Nombre PK :	numfol_mue	
Descripción:	Contiene los registros de muebles tanto de declarante como de cónyuge						
CAMPOS							
Nombre	Tipo y Longitud	PK	FK	CC	Not Null	Default	Descripción
REFEDE_MUE	Texto		X		X		Reg. Fed. del Declarante
REFECA_MUE	Texto		X		X		Registro Federal del Cónyuge
NUMFOL_MUE	Texto	X			X		Número de Folio
TIPDEC_MUE	Texto				X		Tipo de Declaración
TIDECO_MUE	Texto				X		Tipo de Dependiente
TIPMUE_MUE	Texto				X		Tipo de Mueble
DESCRI_MUE	Texto				X		Descripción del Mueble
VALORA_MUE	Numérico				X		Valor
FECREG_MUE	Fecha				X		Fecha de Registro
FEDADQ_MUE	Fecha				X		Fecha de Adquisición
TIPOPER_MUE	Texto				X		Tipo de Operación

5.3.2 Diccionario de Clases

Pagos:

Esta clase registrará pagos como colegiaturas, doctor, farmacia, y cualquiera que no sea una compra de bienes inmuebles o bienes muebles o pagos a inversiones.

Dependencia:

Contendrá un catálogo de Dependencias

Departamento:

Contendrá un catálogo de Departamentos

Declarante:

Contendrá los datos personales del Servidor

Cónyuge:

Contendrá los datos personales de los dependientes o cónyuge del servidor

Tipo de Declaración:

Contendrá el registro de las declaraciones que ha realizado el servidor

Bien Inmueble:

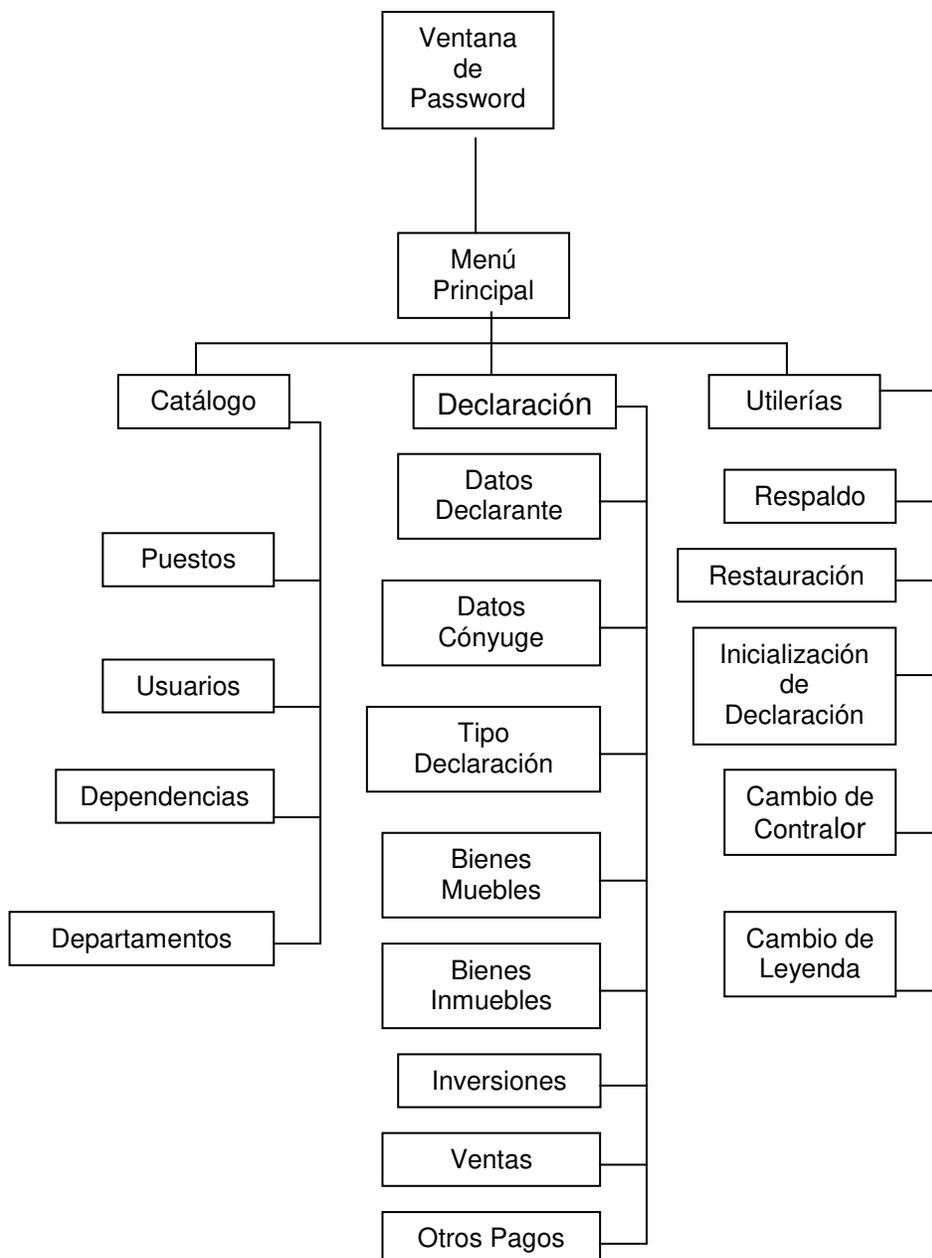
Contendrá la información de los Bienes inmuebles que el servidor y sus dependientes tienen registrados como propietarios.

Bien Mueble:

Contendrá la información de los Bienes muebles que el servidor y sus dependientes tienen registrados como propietarios.

5.3 Interfaz de usuario: Rutas de Navegación y Maquetas de -Pantalla

5.3.1 Ruta de Navegación:



5.3.2 A continuación se muestran algunas pantallas del sistema

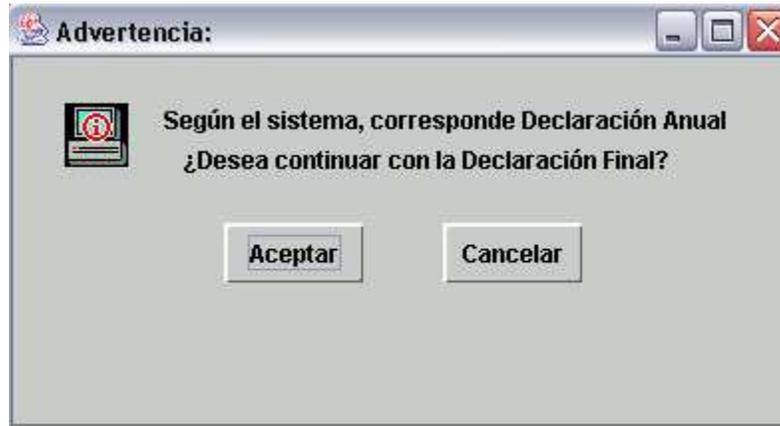
Pantalla de password para entrar al sistema



Pantalla del Menú principal



Pantallas de Advertencia



Pantalla para cálculo, consulta e impresión de la balanza

Balanza General

R.F.C. Declarante: Fecha de Declaración: Tipo de Declaración: Período:

Concepto	Declarante	Cónyuge y/o Dependiente	Total
1. Sueldos, honorarios y otras prestaciones (Bono, aguinaldo, compensaciones)	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
1B. Honorarios	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
2. Rentas, regalías, intereses y dividendos	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
3. Donaciones, herencias, intereses o legados	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
4. Préstamos hipotecarios, personales u otros	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
5. Venta de bienes inmuebles	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
6. Venta de bienes muebles y/o semovientes	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
7. Retiro de ahorro, valores, acciones, etc.	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
TOTAL DE INGRESOS	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
1. Pago de adeudos	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
2. Adquisición de bienes inmuebles	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
3. Adquisición de muebles y/o semovientes	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
4. Saldo en bancos, valores, acciones u otros	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
5. Gastos generales	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
TOTAL DE APLICACIÓN	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

6. Glosario de Términos:

- 6.1 IEEE 830: Estándar para evaluar la especificación de los requerimientos del software
- 6.2 ISO/IEC 12207: Estándar para evaluar las fases del ciclo de vida del software
- 6.3 Requerimiento: una característica que debe tener el sistema o una restricción que debe satisfacer para que sea aceptado por el cliente
- 6.4 Documento de Especificación de Requerimientos: Documento que contiene las especificaciones de los requerimientos del nuevo sistema, que se entrega al cliente
- 6.5 Modelado del Sistema: Representación de entradas, procesos, salidas.
- 6.6 Modelado: Permite al analista crear una jerarquía a detalle.