

DISEÑO DE UNA METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE
COMO HERRAMIENTA DE SOPORTE PARA LA GESTIÓN DE TI

LINA RIOS URUETA
LEONIDAS JIMENEZ GUZMAN

UNIVERSIDAD DEL NORTE
UNIVERSIDAD TECNOLÓGICA DE BOLIVAR
ESPECIALIZACIÓN EN GERENCIA DE SISTEMAS DE INFORMACION
CARTAGENA 2007

DISEÑO DE UNA METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE
COMO HERRAMIENTA DE SOPORTE PARA LA GESTIÓN DE TI

LINA RIOS URUETA
LEONIDAS JIMENEZ GUZMAN

Trabajo Presentado Para
Optar al Título de
Especialistas en Gerencia
de sistemas de Información

Asesor
Adriana Vélez

UNIVERSIDAD DEL NORTE
UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR
ESPECIALIZACIÓN EN GERENCIA DE SISTEMAS DE INFORMACIÓN
CARTAGENA 2007

TABLA DE CONTENIDO

1.	Introduccion	3
1.1.	Introducción	7
1.2.	Descripción del contenido del trabajo integrador	8
2.	Contexto y justificacion	9
3.	Objetivos	12
3.1.	General	12
3.2.	Específicos.....	12
4.	Metodología	14
5.	Diagnostico empresarial.....	15
6.	Conceptualización.....	18
6.1.	Ingeniería del software.....	18
6.2.	Proceso de desarrollo de software.....	20
6.3.	Ciclo de vida del software	21
6.4.	Capability maturity model integration - cmmi	25
6.5.	Rup (rational unified process)	28
6.6.	IEEE standard for developing a software project life cycle process (IEEE std 1074 -2006).....	33
7.	Diseño de la guia metodologica	45
7.1.	Modelo del ciclo de vida.....	45
7.2.	Fases del ciclo de vida para proyectos de software.....	48
7.3.	Actividades y entregables de las fases del ciclo de vida.....	51
7.4.	Actividades de las fases del ciclo de vida	53
8.	Conclusiones	66
8.1.	Aportes.....	67
8.2.	Trabajos futuros	68
9.	Bibliografía y referencias.....	69
	Anexos	70

Anexo A	71
Encuesta para el diagnostico de metodologia de desarrollo de software	71
Anexo B	74
Mapeo de actividades del estándar ieee 1074-1006 en las fases del ciclo de vida	74
Anexo C	76
Lista de actividades del estándar ieee 1074-1006 frete a las actividades pertenecientes a las fases del ciclo de vida	76
Anexo D	79
<u> </u> Formatos de la guía metodológica	79

TABLA DE FIGURAS

Figura 1 Ciclo de vida genérico (PMBOK 2000)4	22
Figura 2. Ejemplo de ciclo lineal para un proyecto de desarrollo	24
Figura 3. Modelo de Ciclo de Vida RUP	29
Figura 4. Modelos de diagramas UML	31
Figura 5. Ejemplo de Casos de Uso	31
Figura 6. Elementos del proceso del ciclo de vida del proyecto de software	40
Figura 7. Flujo de Información en las actividades	41
Figura 8. Ciclo Propuesto Para el Desarrollo de un Proyecto de Software	47
Figura 9. Ciclo Administrativo Propuesto para un Proyecto de Software	48

TABLA DE TABLAS

Tabla 1. Secciones y grupos de actividades	36
Tabla 2. Fases, actividades y entregables de la guía	52

1. INTRODUCCION

1.1. Introducción

Contar con una metodología sencilla, de bajo costo de implementación, fácil de comprender y ajustable a las necesidades del área de desarrollo, es una de las preocupaciones de los departamentos de tecnología.

Pero elaborar una guía demanda conocimiento en los estándares existentes, capacidad de análisis y experiencias en la aplicación de metodologías. Por estas razones los gerentes de tecnología en especial en empresas pequeñas y medianas de nuestro entorno no invierten esfuerzo en su búsqueda, estudio e implementación, ya que su tiempo se consume en el día a día y en entregables que se producen sin ningún tipo de planeación y consideraciones.

Escoger una metodología es una tarea clave para un gerente de tecnología o de proyecto, porque de ella se desprende el nivel de confianza y de compromiso del grupo y la escogencia de las actividades de cada fase del ciclo es decisiva para la buena marcha del proyecto.

Por lo anterior, las metodologías más que ofrecer un paso a paso buscan abrir un panorama de actividades que se usan o no dependiendo del proyecto y que pueden crecer en insumos, características y controles.

La implementación de metodologías basadas en estándar permite la aplicación de mejores prácticas dentro de la organización.

Lo anterior sugiere que existe la necesidad de una guía que:

- Asista al responsable de proyecto en la administración y control de proyectos.
- Permita estandarizar los procesos del área de desarrollo.
- Ponga al servicio de la organización el conocimiento adquirido en el desarrollo de proyectos.

El presente trabajo presenta una guía para el desarrollo de proyectos de software aplicando el enfoque del estándar IEEE 1074 -2006 standard for developing a software project life cycle process.

1.2. Descripción del contenido del trabajo integrador

El trabajo integrador consta de siete capítulos:

- Capítulo 2, hace una descripción de la situación actual que enfrentan los departamentos de tecnologías en empresas colombianas categorizadas como pequeñas y medianas haciendo énfasis en los problemas que afrontan en gran medida como consecuencia de la falta de planeación, metodologías y controles.
- Capítulo 3, describe los objetivos a alcanzar en el desarrollo de este trabajo.
- Capítulo 4, expone la metodología a seguir para el desarrollo del trabajo.
- Capítulo 5, detalla los resultados obtenidos en la encuesta realizada a un grupo de empresas colombianas permitiendo conocer la situación con respecto a estándares y metodologías.
- Capítulo 6, describe la fundamentación teórica que sirve de marco para el desarrollo de la guía metodología para proyectos de software
- Capítulo 7, describe la guía propuesta por el grupo de trabajo, La guía presentan los pasos a seguir para su desarrollo y los formatos que servirán de soporte durante el proyecto.

2. CONTEXTO Y JUSTIFICACION

El comportamiento recurrente de las áreas de TI al enfrentar día a día las actividades sin contar con herramientas procedimentales y computacionales que soporten las tareas de planeación, administración y control, ha ocasionado la carencia de procedimientos, estándares y datos estadísticos que les permita iniciar procesos de mejora al interior del área, así como tareas encaminadas al conocimiento y análisis de su entorno. Alcanzar este grado de madurez es lo que permitirá que el área de TI sea considerada como "*aliado estratégico*" en las organizaciones y no simplemente como una unidad de soporte.

El mundo de la informática esta direccionándose hacia el establecimiento de estándares que permitan, a las organizaciones, tanto la unificación de criterios, como la adopción de políticas y estrategias para alcanzar un grado de madurez que esté acorde con su entorno.

Adicionalmente, en los últimos años han entrado a jugar como factor importante los conceptos de mejoramiento de procesos, aseguramiento de calidad y mejores prácticas, los cuales persiguen obtener provecho del conocimiento y la experiencia de otros para poder aplicarlo en el ambiente de trabajo, extrayendo lo mejor y adecuándolo a las condiciones de la organización.

Estudiando las estrategias y metodologías utilizadas en las pequeñas empresas Colombianas, que cuentan con un área de TI cuya misión es dar soporte y desarrollar soluciones a la medida de las necesidades que se presenten, se observa que en la mayoría de los casos los proyectos de software se realizan sin tener una metodología adecuada para la planeación, control y seguimiento de las actividades.

Esto obedece a la errónea concepción que la implementación de metodologías, estándares y herramientas de apoyo conlleva altos costos tanto en recurso humano como en tecnología; sin embargo, no se detienen a considerar los grandes beneficios que se obtienen al aplicarla, dando así solución a problemas tangibles y potenciales, tales como:

- Documentación incompleta o inexistentes
- Falta de control de fuentes y versiones
- Altos costos en la etapa de mantenimiento debido al gran número de cambios requeridos después de la entrega
- Los programadores inician la etapa de desarrollo sin tener un diseño completo de la aplicación, y en ocasiones sin contar con las especificaciones completas
- El producto entregado no se acerca a las necesidades del usuario
- Falta de estándares y metodología de trabajo en el grupo de desarrolladores
- Pruebas insuficientes
- Levantamientos de información imprecisos

Los motivos anteriormente mencionados, sugieren la necesidad de definición de una guía metodológica que:

- Asista al responsable en el proceso de planeación, seguimiento y control de los proyectos de desarrollo de software.
- Unifique los procesos que conlleva el desarrollo del proyecto, ofreciendo una metodología que permita a los integrantes del grupo de trabajo llevar el mismo ritmo en la ejecución de este.
- Se convierta en una buena práctica para el equipo de trabajo, con el fin de mantener los estándares de calidad para alcanzar proyectos exitosos.

- Contribuya a la profesionalización de las áreas de desarrollo de TI en el ambiente colombiano

3. OBJETIVOS

3.1. General

Diseñar una guía metodológica que le permita a las áreas de TI planear y administrar el desarrollo de proyectos de software, manteniendo el flujo del proceso con el fin de tener una adecuada:

- Documentación
- Administración de recursos y apoyo de la dirección
- Disminución de costos
- Administración dinámica de los proyectos, con el fin de tener el proyecto en marcha y finalizarlo de forma efectiva
- Delimitación y planeación del proyecto
- Profesionalización del trabajo
- Estandarización de procesos
- Bases de conocimientos
- Administración activa del riesgo

3.2. Específicos

Conocer los estándares metodológicos más representativos existentes para la planeación y administración de desarrollo de software.

Adquirir y estudiar el estándar IEEE 1074-2006 IEEE Standard for Developing a Software Project Life Cycle Process, como guía para el desarrollo de la metodología propuesta

Realizar un diagnostico de la situación de los áreas de desarrollo en TI para tener una aproximación del estado real en empresas pequeñas y medianas en Colombia

Preparar la metodología acorde con el proceso del ciclo de vida de proyectos de software

Concluir y presentar propuestas de mejoras para trabajos futuros basadas en el presente proyecto

4. METODOLOGÍA

Las fases y etapas de esta metodología, que servirán de directriz en el desarrollo de la guía metodológica, son las siguientes:

Fase I – Adquisición de requisitos

La adquisición y estudio del estándar IEEE 1074-2006.

Fuentes de información necesarias para la preparación de la guía metodológica

Fase II- Diagnostico empresarial

- Preparación de encuestas
- Aplicación de encuestas en las empresas colombianas seleccionadas como muestra
- Presentación de resultados

Fase III- Conceptualización

Esta fase permite, comprender el ámbito del problema y la terminología aplicada.

Fase IV – Diseño de la guía metodológica

Basado en la conceptualización y en el estándar de la IEEE 1074 modelar una guía metodológica para soportar el proceso del ciclo de vida de los proyectos de desarrollo de software, que sea aplicable en las áreas de desarrollo de TI, en empresas pequeñas y medianas.

5. DIAGNOSTICO EMPRESARIAL

Con el fin de conocer la situación al interior del Área de Tecnología de las organizaciones pequeñas y medianas en Colombia, referente a la aplicación de metodologías en la gestión de proyectos de desarrollo de software, se elaboró una encuesta (ver Anexo A) que arrojó los siguientes resultados.

La encuesta fue aplicada en siete empresas colombianas: El Colombiano, Vanguardia Liberal, El Universal, Sociedad Portuaria de Cartagena, Autobol, Universidad Tecnológico de Confenalco y LinkedIP.

Los resultados fueron los siguientes:

El 100% de las empresas cuentan con área de tecnología.

El 71% de las empresas tienen definidos la misión y la visión del área de TI y de este grupo el 86% la misión y visión están alineadas con la estrategia de la organización.

El 43% de las áreas de TI desarrollan planes estratégicos anualmente y 86% de las empresas encuestadas cuentan con áreas de TI que tienen definidas las metas y recursos para proyectos de desarrollo de software.

86% de las áreas de TI tienen área de desarrollo, con cinco personas en promedio. Solo el 43% tienen personas dedicadas a gerenciar los proyectos del área.

El 71% de las empresas encuestadas considera que es importante la implementación de una metodología de desarrollo de software, el 43% de las áreas de TI tienen implementada una metodología para el ciclo de vida de los proyectos de desarrollo de software y solo el 29% de este grupo que corresponde a dos empresas, aplican un estándar en la metodología y siguen el ciclo de vida estándar: Análisis, diseño, desarrollo, pruebas e instalación y seguimiento. Una de las empresas, aplica el estándar Rational Unified Process RUP, siguiendo sus mejores practicas, y la otra sigue la programación paralela.

El 100% de las empresas reconoce la importancia de implementar una metodología para gestionar los proyectos de software pues consideran que no solo ayudan para el control del ciclo de vida en cuanto a sus fases de Análisis – Diseño – Desarrollo - Pruebas – Instalación – Seguimiento, sino que apuntan a un adecuado seguimiento a los cambios, favoreciendo lo que a futuro continua al instalar un software y es su mantenimiento y acople a los cambios inherentes en el entorno.

Adicionalmente, disciplina en el buen manejo de los formatos que complementan el desarrollo tales como: Requerimientos, casos de uso, flujo de actividades, diagramas de flujo, modelo de datos, formatos de seguimiento a cambios, etc., que apoyan y complementan sustancialmente la documentación y la posibilidad que otro analista pueda intervenir con su desarrollo en el mismo sistema sin mayores tropiezos de entendimiento de lo que otro ya realizó.

El 57% de las áreas de TI en las empresas encuestadas tienen control de versiones y fuentes y el tiempo dedicado a mantenimientos en promedio es del 20%.

El 43% de las áreas de TI de la muestra, cuenta con procedimiento y formatos para el levantamiento de requerimientos.

El 57% de la muestra documenta sus nuevos desarrollos y el 43% de la muestra documenta los mantenimientos.

A la pregunta ¿Cual es la mayor problemática que afrontan las áreas de TI?, estas fueran las respuestas:

- Desarrollos imprevistos que demandan prontitud y en la planeación de éstos no se involucra el área de tecnología
- La carencia de una metodología para el desarrollo de proyectos de Software. Falta de estándares y documentación de los mismos
- Dificultad en la implementación de estándares y metodologías debido a la cultura organizacional
- Falta de manejo de fuentes y versiones
- Personal de desarrollo en labores de soporte.
- Variabilidad de los requerimientos.
- Adaptación a las nuevas tecnologías de desarrollo
- Falta de compromiso en la ejecución de tareas a cargo de otras áreas diferentes a TI
- Relaciones con el proveedor del producto en aquellas donde se da Outsourcing

6. CONCEPTUALIZACIÓN

6.1. Ingeniería del software

En el glosario estándar de terminología de IEEE (IEEE Std 610.12-1990) ¹ se define Ingeniería de Software como “la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software”.

Fritz Bauer, definió la Ingeniería de Software como “el establecimiento y uso de principios de ingeniería robustos, orientados a obtener económicamente software que sea viable y funciones eficientemente sobre maquinas reales”²

La ingeniería de software es un esquema metodológico que permite producir programas para computador tendiente a satisfacer una necesidad, incluyendo los mecanismos necesarios de mantenimiento encaminados a soportar su operación. La ingeniería del Software, en si misma, ofrece herramientas para soportar cada uno de los momentos de la vida del sistema incluyendo las etapas del ciclo del desarrollo, más que ser un compendio de técnicas de programación.

La Ingeniería del Software ha definido las etapas que son requeridas para llevar a cabo el desarrollo de los programas, y por cada una de estas etapas se han diseñado estándares de definición, planeación y control, que permiten hacer del Ciclo de Vida del Software una actividad medible y procedimental.

¹ IEEE. IEEE Std 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology. IEEE, 1990.

La ingeniería del software trata con áreas muy diversas de la Informática y de las Ciencias de la Computación, tales como construcción de compiladores, sistemas

² PRESSMAN, Roger. Software Ingeneering, a Practitioner's Approach. Sexta Edición. McGraw-Hill Profesional. 2005.

operativos o desarrollos de Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a una infinidad de áreas tales como: negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, el mundo del derecho, la red de redes Internet, redes Intranet y Extranet, etc.

Una perspectiva industrial

En los inicios de la informática, los sistemas basados en computador se desarrollaban usando técnicas de gestión orientadas a hardware. Los gestores de proyecto se centraban en el hardware, debido a que era el factor principal del presupuesto en el desarrollo del sistema. Para controlar los costos del hardware, los gestores establecieron controles formales y estándares técnicos. Exigían un análisis y diseño completo antes de que algo se construyera. Medían el proceso para determinar donde podía hacerse mejoras. En resumen, aplicaban los controles, los métodos y las herramientas que reconocemos como Ingeniería del Hardware, la programación se veía como un arte. Existían pocos métodos formales y pocas personas los usaban.

Hoy, la distribución de los costos en el desarrollo de sistemas informáticos ha cambiado drásticamente. El software, en lugar del hardware, es el elemento principal del costo. Por esto, los gerentes de proyecto y técnicos se hacen preguntas relacionadas con el tiempo de construcción de los programas, el costo, la calidad del software y la medición del progreso del desarrollo de software y para esto, han adoptado la Ingeniería de Software como practica.

6.2. Proceso de desarrollo de software

El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" ³.

El proceso de desarrollo de software es el conjunto de técnicas y procedimientos que nos permiten conocer los elementos necesarios para definir un proyecto de software.

La IEEE clasifica los procesos de desarrollo de software en funcionales y no funcionales:

6.2.1. Funcionales

Condición o capacidad, de un sistema, requerida por el usuario para resolver un problema o alcanzar un objetivo.

6.2.2. No Funcionales

Condición o capacidad que debe poseer un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto.

6.3. Ciclo de vida del software

Crear un producto exitoso requiere identificar las necesidades del mercado y trasladarlas, dentro de la visión de un producto y un enfoque, a que sean ejecutadas siguiendo las pautas de los principios de administración de proyectos.

³ JACOBSON. Disponible en <http://www.angelfire.com/scifi/jzavalar/apuntes/#Jaco-bson>

Como los proyectos son únicos, estos generan un grado de incertidumbre. Las organizaciones ejecutan proyectos que usualmente son divididos en fases con el fin de mejorar su administración y control, logrando proveer enlaces a las operación repetitivas de la organización. A estas fases se les conoce como ciclo de vida de los proyectos.(PMBOOK 2000)⁴

El ciclo de vida del software es la suma de todas las actividades necesarias para definir, desarrollar, implementar, construir, operar, mantener y desmontar un producto y las variantes asociadas a él.

El ciclo de vida es la representación de las fases a través de las cuales pasa un proyecto, definiendo su inicio y fin.

- Cada fase del proyecto produce uno o más entregables.
- Cada fase genera un punto de evaluación

La administración del proyecto es el rol de gobernar un producto desde el inicio hasta generar el mayor valor agregado posible al negocio. Los requerimientos son los bloques de construcción básicos que unen las diferentes fases del ciclo de vida del producto

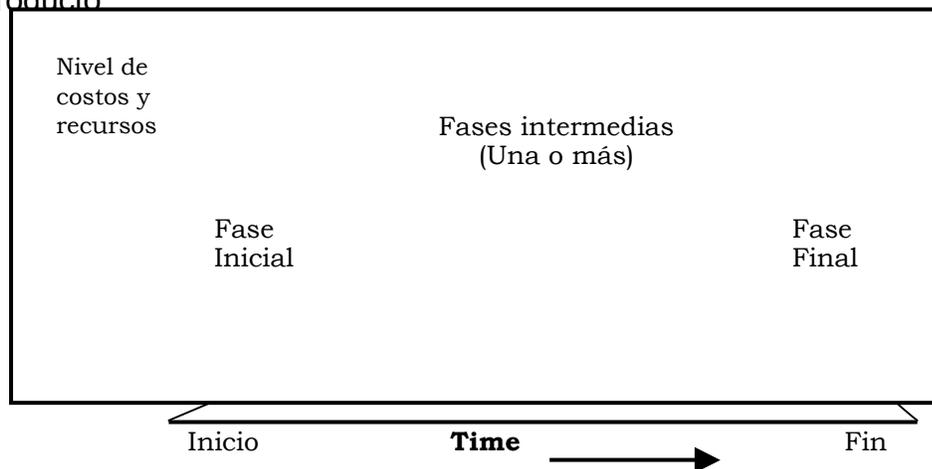


Figura 1 Ciclo de vida genérico (PMBOOK 2000)⁴

⁴ PROJECT Management Institute, Project Management body of knowledge, PMBOOK Guide, Newtown Square, Pennsylvania USA, Edition 2000

Muchos ciclos de vida de proyectos tienen fases similares, con entregables similares, pero pocos son idénticos. Muchos tienen 4 ó 5 fases, pero algunos tienen 9 ó más fases.

Dentro de un área de desarrollo básica, puede haber variaciones significativas, un ciclo de vida de desarrollo de software en una organización puede tener una fase de diseño, mientras que otro ciclo puede tener fases separadas para el diseño funcional y el diseño de detalle.

La descomposición temporal en 'fases' o 'etapas' del ciclo busca:

- Mejor control de la gestión,
- Adecuados enlaces con las operaciones habituales de la organización.
- Cada fase supone la realización completa de uno o varios 'entregables'
- Un entregable es un producto tangible y comprobable; por ejemplo, un estudio de viabilidad, un diseño detallado o un prototipo.
- La conclusión de una fase supone revisar el entregable y la ejecución del proyecto con el fin de determinar si el proyecto debe continuar y detectar y corregir errores con un costo asumible.

Para cada fase se define:

- Su secuenciación temporal respecto de las demás fases.
- El trabajo que debe realizarse (incluidos entregables), y
- Los participantes.

Existen muchos modelos de ciclos de vida.

6.3.1. Modelo de ciclo de vida lineal

Es el más utilizado, siempre que es posible, precisamente por ser el más sencillo. Consiste en descomponer la actividad global del proyecto en fases que se suceden de manera lineal, es decir, cada una se realiza una sola vez, cada una se realiza tras la anterior y antes que la siguiente. Con un ciclo lineal es fácil dividir las tareas entre equipos sucesivos, y prever los tiempos (sumando los de cada fase).

Requiere que la actividad del proyecto pueda descomponerse de manera que una fase no necesite resultados de las siguientes (realimentación), aunque pueden admitirse ciertos supuestos de realimentación correctiva.

Desde el punto de vista de la gestión (para decisiones de planificación), requiere también que se sepa bien, de antemano, lo que va a ocurrir en cada fase antes de empezarla.

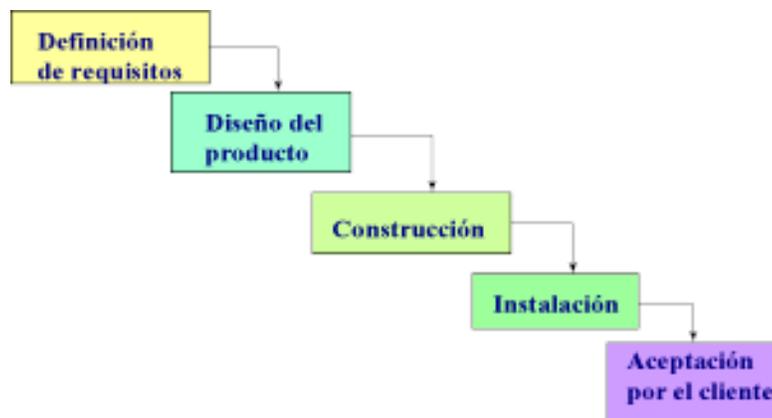


Figura 2. Ejemplo de ciclo lineal para un proyecto de desarrollo

6.3.2. Modelo de ciclo de vida de software en espiral (por Muench)

Muench describe un modelo de ciclo de desarrollo de software en cuatro cuadrantes y en cuatro ciclos.

Los cuadrantes son:

- Definición
- Diseño
- Construcción
- Evaluación

Los ciclos son:

- Ciclo de prueba del concepto: contiene los requerimientos del negocio, define las metas para la prueba de concepto, produce la aceptación de los planes de prueba, conduce los análisis de riesgos y realiza recomendaciones
- Ciclo de primera construcción: obtiene los requerimientos del sistema, define las metas para la primera construcción, produce el diseño del sistema lógico, diseña y construye la primera construcción, produce los planes de prueba del sistema, evalúa la primera construcción, y realiza recomendaciones.
- Ciclo de segunda construcción: obtiene los requerimientos de los subsistemas, define las metas para la segunda construcción, produce el diseño físico, construye la segunda construcción, produce los planes de prueba del sistema, evalúa la segunda construcción, y realiza recomendaciones
- Ciclo final: completa los requerimientos de las unidades, diseño final, realiza la construcción final, las pruebas de aceptación de la unidad, del subsistema y del sistema

6.4. Capability maturity model integration - CMMI

CMMI (Capability Maturity Model Integration) es un conjunto de modelos que permite a las organizaciones medir e incorporar mayores niveles de eficacia y madurez en sus procesos de desarrollo y mantenimiento de software, y está considerado como uno de los mayores referentes mundiales en cuanto a producción de software.

La incorporación de esta metodología de trabajo supone una sólida apuesta por las mejores prácticas de gestión de la tecnología a nivel mundial.

El CMMI establece cinco niveles progresivos para los procesos relacionados con la construcción de aplicaciones informáticas, hasta alcanzar la madurez total:

6.4.1. Inicial

Está caracterizado por una aproximación intuitiva al proceso de desarrollo del software. El éxito depende del esfuerzo individual. No se han definido procesos metodológicos, o se han definido pero no se siguen. Es necesario realizar medidas de línea base, es decir, medidas que servirán para estimar y planificar en el futuro. Asimismo, es el momento de hacer un esfuerzo de estructuración y control en el proceso.

6.4.2. Repetible

La madurez metodológica de la organización permite estimar fiablemente el tamaño funcional o físico del sistema, así como recursos, esfuerzo, costos y calendario. Se han sentado las bases para repetir éxitos anteriores en proyectos con aplicaciones similares.

Las áreas clave de proceso definidas en este nivel, cuyo estado se puede conocer mediante diversas métricas, son las siguientes:

- Gestión de requisitos.
- Planificación del proyecto software.
- Seguimiento y control del proyecto.
- Gestión de la subcontratación del software.
- Aseguramiento de la calidad del software.
- Gestión de la configuración del software.

6.4.3. Definido

Se conoce la forma de construcción del sistema. El proceso del software de las actividades de gestión e ingeniería se documenta y se estandariza. Las actividades intermedias están bien definidas, y por tanto se pueden examinar y medir.

Las áreas clave definidas en este nivel son:

- Desarrollo y mejora de los procesos de la organización.
- Definición de los procesos de la organización.
- Programa de formación.
- Gestión integrada del software.
- Ingeniería de producto software.
- Coordinación Intergrupos.
- Revisión conjunta.

6.4.4. Gestionado

Se añade la gestión a un proceso definido. Se usa realimentación desde las primeras actividades del proyecto para seleccionar prioridades en las actividades actuales y conocer cómo se emplean los recursos. Los efectos de los cambios en

una actividad se pueden seguir en otras. Se recopilan medidas detalladas del proceso del software y de la calidad del producto. En definitiva, se evalúa la efectividad de las actividades del proceso. Por ejemplo, se podría medir cuánto se está produciendo para ser reutilizado, cuánto se está reutilizando de proyectos anteriores, cómo y cuándo son descubiertos los defectos y la relación entre fechas de finalización de los módulos y fechas previstas.

Las áreas clave definidas en este nivel son dos:

- Gestión cuantitativa del proyecto.
- Gestión de calidad del software.

6.4.5. Optimizado

Existe una mejora continua de los procesos. Las medidas de actividades se usan para mejorar el proceso, eliminando y añadiendo actividades y reorganizando su estructura como respuesta a los resultados de las medidas.

Las áreas definidas para este nivel son:

- Prevención de defectos.
- Gestión de cambios tecnológicos
- Gestión de cambios en los procesos.

6.5. RUP (RATIONAL UNIFIED PROCESS)

El Racional Unified Process es una extensión del concepto de Proceso Unificado de Desarrollo de Software o simplemente del Proceso unificado. Sus orígenes se remontan al modelo de Barry Boehm que hace referencia al modelo en espiral. El RUP en si no es un sistema con pasos preestablecidos, sino por el contrario es un

conjunto de metodologías que se pueden adaptar de acuerdo a las necesidades y al contexto de cada organización. Una de las principales características de la metodología es que se implementa de forma iterativa e incremental, asistida por los Casos de Usos, definidos por el UML (Lenguaje Unificado de Modelado), brindando un marco de desarrollo. Dichos casos de uso permiten identificar los requerimientos funcionales del sistema en desarrollo.

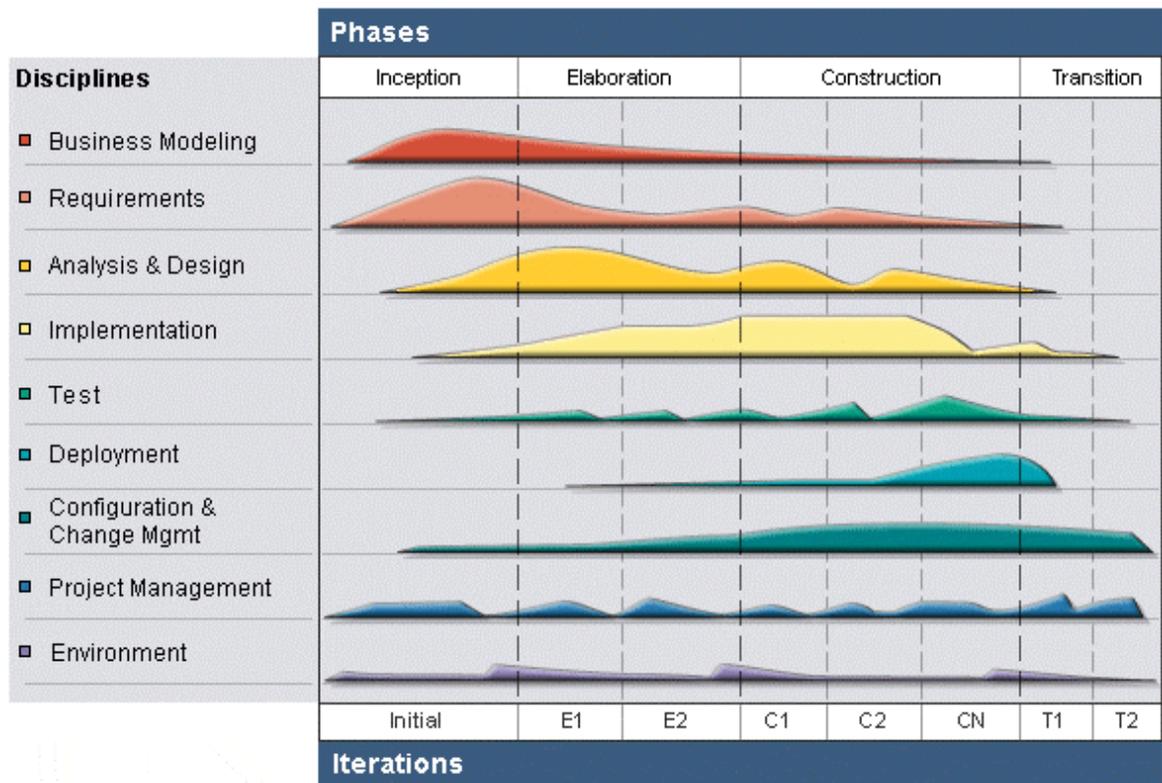


Figura 3. Modelo de Ciclo de Vida RUP⁵

Tal y como se ve en la Figura 3, en un proyecto desarrollado mediante RUP se identifican cuatro fases:

- Inicio
- Elaboración

- Construcción
- Transición

Cada una de dichas fases esta dividida en iteraciones (Excluyendo la fase de Inicio que solo tiene iteraciones en casos de proyectos muy grandes). Al finalizar cada una de las fases se cuenta con incremento del producto, entregable y que cuenta con adiciones o mejoras a la funcionalidad del sistema que se esta desarrollando.

Por su parte en cada una de las iteraciones se deben realizar actividades de:

- Modelado del negocio
- Definición de requerimientos
- Análisis y diseño
- Implementación
- Pruebas
- Despliegue
- Gestión de configuración y cambios
- Gestión del proyecto
- Entorno

Estas actividades son similares a las actividades realizadas en el ciclo de vida del desarrollo de software tradicional.

⁵ LETELIER, Patricio. Introducción a Rational Unified Process (RUP). Disponible en www.dsic.upv.es/~letelier/pub

Workflow:Requirimientos

Workflow Detail: Analizar el Problema

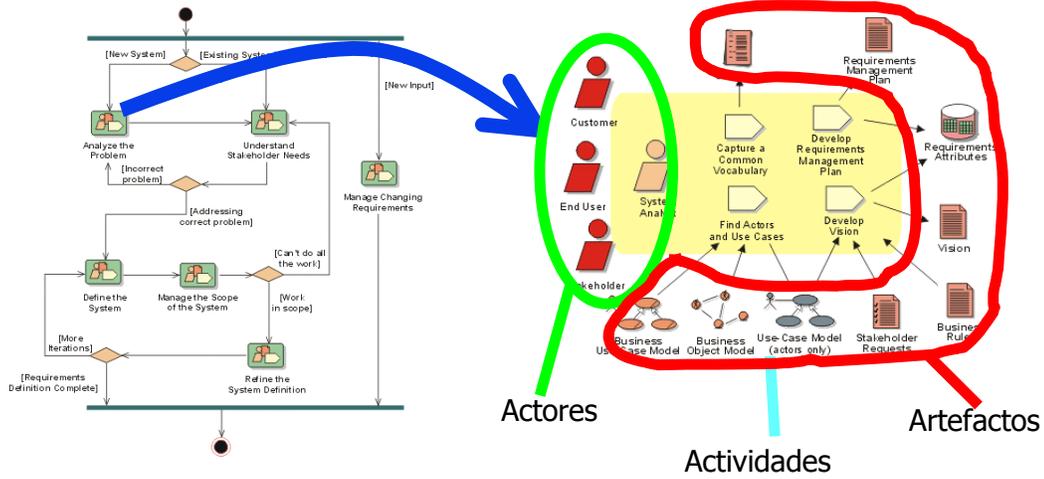


Figura 4. Modelos de diagramas UML⁶

Mediante la utilización de UML se define el Workflow del proceso, posteriormente se identifican los componentes para llegar a los casos de uso.

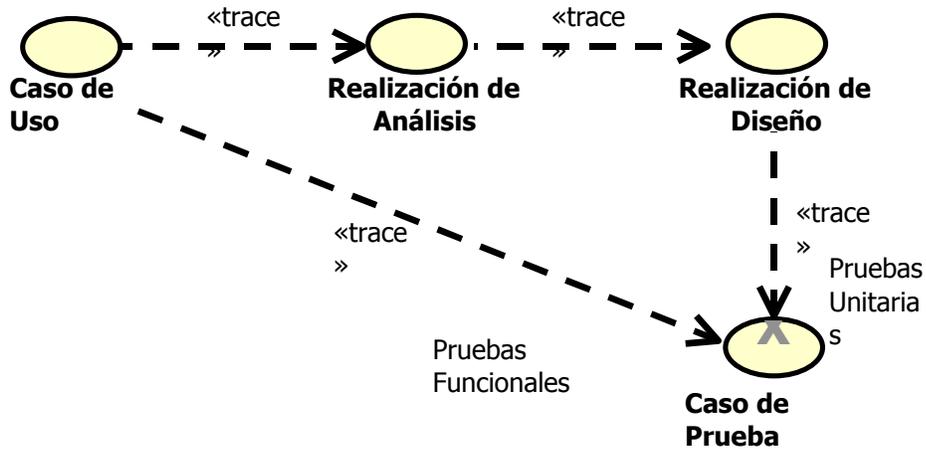


Figura 5. Ejemplo de Casos de Uso⁷

La Figura 5 muestra un caso de uso típico.

⁶ LETELIER, Patricio: [Introducción a Rational Unified Process \(RUP\)](http://www.dsic.upv.es/~letelier/pub). Disponible en internet: www.dsic.upv.es/~letelier/pub

⁷ LETELIER, Patricio: [Introducción a Rational Unified Process \(RUP\)](http://www.dsic.upv.es/~letelier/pub). Disponible en internet: www.dsic.upv.es/~letelier/pub

Una de las fortalezas del método es que se orienta a la arquitectura, ya que no existe un modelo único que se aplica a todos los proyectos, por el contrario, para cada proyecto en particular se van definiendo los modelos que determinan la arquitectura del sistema. A su vez esta metodología permite determinar los riesgos en cada una de las etapas, de tal forma que se puedan tener en cuenta y abordar durante la ejecución; facilitando que durante la etapa de elaboración se puedan organizar los riesgos y resolverlos en orden de importancia.

Los principios claves que rigen al RUP son:

- Adaptación del proceso. Es decir que el proyecto debe adaptarse a las características reales del proceso y la organización.
- Balanceo de Prioridades. Como se tiene un equipo de trabajo y cada uno de ellos tiene establecidas prioridades y existen requerimientos mismos del desarrollo, es necesario que se nivelen las cargas para mantener al equipo homogéneo.
- Colaboración entre el equipo. Es de vital importancia que exista mucha comunicación entre los miembros del equipo de desarrollo y de este con su medio ambiente, de tal forma que se puedan coordinar con facilidad actividades, planes, requerimientos etc.
- Valor agregado en cada iteración. Cada vez que se esta en una iteración es necesario verificar la calidad, los aportes al sistema y la estabilidad del producto ya que una salida de cada iteración es el producto ampliado o mejorado.
- Nivel de abstracción. Aunque este concepto parece muy técnico, lo que se pretende es crear la conciencia en el equipo de trabajo de reutilización de código, adecuación de mejores practicas, utilización de Frameworks, facilitando que los programadores no solamente escriban código, sino que por el contrario siempre estén dispuestos a pensar en las implicaciones y requerimientos de lo que están realizando.

- Enfoque en la calidad. Durante todo el proceso de desarrollo se debe estar controlando la calidad del producto con el fin que al finalizar el proyecto estén consideradas todas las condiciones necesarias que garanticen la calidad.

6.6. IEEE standard for developing a software project life cycle process (IEEE STD 1074 -2006)

El estándar IEEE 1074-2006, es la última versión del estándar definido por The Institute of Electrical and Electronics Engineers IEEE, el cual está orientado a permitir establecer un proceso para la administración y mantenimiento de proyectos de desarrollo de software.

El estándar le permite al usuario, luego de escoger el Modelo del Ciclo de Vida del Proyecto de Software, el cual va a aplicar en su proyecto y que debe estar de acuerdo a la misión, visión, metas y recursos de la organización, establecer una serie de actividades que se deben realizar durante la ejecución del desarrollo, el resultado de escoger el modelo y determinar las actividades a realizar dentro del Ciclo de Vida es el Proceso del Ciclo de Vida del Proyecto de Software.

De igual forma el estándar claramente indica que no pretende definir un Ciclo de Vida del Proyecto de Software (SPLC), ni tampoco definir un modelo SPLCM específico, básicamente está orientado a servir de guía en la lista de actividades que se deben contemplar. Sin embargo, si se puede utilizar para desarrollar un proceso organizacional que soporte el desarrollo y mantenimiento de proyectos de software.

El estándar define actividades que están orientadas a cada una de las etapas del ciclo de vida de los proyectos de software incluyendo desde su concepción, desarrollo hasta su retiro o el momento en el que se da de baja al software. La

forma en la que esta concebido el estándar le da mucha flexibilidad en su aplicación, ya que las actividades cubren una amplia gama de condiciones y el usuario determina cuales de ellas debe incluir en cada una de las etapas del Ciclo de Vida del Proyecto de Software (SPLC).

6.6.1. Historia

- En 1995, se realizo la primera revisión de la versión original que se realizo en 1991, durante esta revisión solamente se corrigieron algunos pequeños errores encontrados.
- En 1997 se dieron los siguientes cambios:
 - Las actividades se reordenaron dentro de grupos lógicos los cuales son llamados grupos de actividades.
 - El termino *Grupo de Actividades* entro a reemplazar el termino *Proceso*
 - Se incluyo la actividad denominada Administración del Riesgo
 - Luego de determinar que el software puede ser adquirido en el extranjero, fue necesario incluir el Grupo de Actividades Importación de Software.
- La ultima revisión se realizo en el 2006 y se incluyeron los siguientes cambios:
 - El foco del estándar se centró en un proceso particular de un proyecto determinado.
 - Se incluyeron actividades para la Administración de versiones
 - Por el amplio crecimiento de las necesidades de la seguridad del software se definieron dos actividades nuevas: Determinar Objetivos de Seguridad y Confirmar la Acreditación de la Seguridad.

6.6.2. Metodología

La metodología para el Proceso del Ciclo de Vida del Proyecto de Software (SPLCP) es la siguiente:

- Seleccionar el modelo mas apropiado del ciclo de vida del proyecto de software (SPLCM) que se va a utilizar en el Proceso. Esta escogencia se debe realizar dentro de la lista de modelos definidos, y ojala utilizados, dentro de la organización en sus proyectos anteriores.
- Definir el Ciclo de Vida del Proyecto de Software (SPLC) mediante la utilización del modelo identificado en el paso anterior y la lista de actividades seleccionadas del grupo de actividades incluidas dentro del estándar.
- Crear el SPLCP a partir del SPLC y la inclusión de los valores del proceso organizacional (OPA).

Un punto muy importante es que las actividades que vienen definidas en el estándar, en su totalidad están orientadas al desarrollo de software, por tanto no incluye actividades que no tengan que ver con el proceso, tales como contratación, compra o desarrollo de Hardware. De igual forma el estándar presume que el arquitecto del proceso esta familiarizado con los diferentes SPLCM existentes en la organización, así como debe estar familiarizado con los requerimientos, restricciones, características y medio ambiente, propias del sistema que se va a desarrollar. Finalmente, aunque el estándar define una lista de actividades que se deben realizar, no indica la forma en la cual se deben llevar a cabo.

6.6.3. Grupos de actividades

El estándar organiza y divide el proceso en Secciones y cada una de estas está conformada por grupos de actividades.

Sección	Grupos de Actividades
Administración del Proyecto	Iniciación del Proyecto Planeación del Proyecto Monitoreo y Control del Proyecto
PRE-Desarrollo	Exploración Conceptual Localización del Sistema Importación de Software
Desarrollo	Requerimientos de Software Diseño Implementación
Post-Desarrollo	Instalación Operación y Soporte Mantenimiento Retiro (Dar de Baja)
Soporte	Evaluación Administración de la configuración de Software Documentación Entrenamiento

Tabla 1. Secciones y grupos de actividades

El estándar incluye una lista de 69 actividades, las cuales cubren todo el Ciclo de Vida del Software y se reparten dentro de los 17 grupos de actividades.

6.6.4. Definiciones utilizadas en el estándar

6.6.4.1. (SPLC) Ciclo de Vida del Proyecto de Software

Es la parte del Ciclo de Vida de Software que es aplicable a un proyecto específico. Por tanto es una lista de actividades seleccionadas para el Modelo del

Ciclo de Vida del Proyecto de Software (SPLCM) escogido para el proyecto en particular.

6.6.4.2. (SPLCM) Modelo del Ciclo de Vida del Proyecto de Software

Es el modelo seleccionado para un proyecto en particular, el cual es escogido de la lista de modelos definidos dentro de la organización y que junto con la selección de la lista de actividades determinadas a ser utilizadas en el proyecto forman el SPLC.

6.6.4.3. (SPLCP) Proceso del Ciclo de Vida del Proyecto de Software

Es el proceso definido para un proyecto en particular, el cual es el resultado de la inclusión de los valores del proceso organizacional (OPA's) dentro del Ciclo de Vida del Proyecto de Software (SPLC).

6.6.4.4. Grupo de Actividades de Soporte

son las actividades necesarias para garantizar la finalización exitosa del proceso, pero que no son actividades propiamente dichas del proceso de desarrollo.

6.6.4.5. Modelamiento de la Amenaza

Es una técnica para determinar circunstancias o eventos que pueden constituirse en un detrimento del sistema, tales como, destrucción, modificación de datos o denegación del servicio y que pueden convertirse en un peligro de vulnerabilidad.

6.6.4.6. Acreditación de Seguridad

Es una declaración formal que un sistema de información esta aprobado para trabajar bajo un modelo de seguridad que le da la posibilidad de operar en un nivel de riesgo aceptable.

6.6.4.7. Arquitecto del Proceso

Es la persona o entidad organizacional con la autoridad y responsabilidad para desarrollar y mantener SPLCPs.

6.6.4.8. Valores del proceso organizacional (OPAs)

Son los elementos que define el ambiente en la organización para los proyectos de software. Dichos elementos son seleccionados y adaptados para cada proyecto en particular. Debido a que los OPAs son particulares para cada organización, la colección varía de una compañía a la otra. La recomendación del estándar es que se incluyan dentro de los OPAs políticas, procedimientos, estándares, métricas, metodologías, etc.

6.6.4.9. Mapeo

Esta es la técnica utilizada para establecer la secuencia de tareas ejecutables dentro de un SPLCM. Las actividades se pueden mapear de tres maneras como instancia, como iteración y como invocación.

6.6.4.10. SPMPi

Información de la planeación del manejo del proyecto de software.

6.6.5. Actividades

Según el estándar, las actividades son definidas como cuerpos de trabajo a ser ejecutados, incluyendo la información de entrada y la información de salida que se requiere; explícitamente corresponde a la transformación de la información de entrada en la información de salida. La actividad se da por completada cuando toda la información de entradas ha sido procesada toda la información de salida, que aplica, ha sido generada.

Las actividades están conformadas de tres elementos:

- Información de entrada. Es la lista de información que va a ser procesada y las fuentes que la proveen.
- Descripción: Discusión de las acciones de Valor-Agregado a ser ejecutadas para complementar la Transformación.
- Información de Salida: Lista de la información de salida que se va a generar mediante la transformación de la información de entrada, y los destinatarios de dichas salidas.

Para *comenzar o entrar* a una actividad, al menos un elemento de la información de entrada, esta disponible o presente. Para *Completar o salir* de una actividad, toda la información de entrada ha sido procesada, y toda la información de salida ha sido generada, teniendo en cuenta que es el modelo de Ciclo de Vida seleccionado el que determina que información es pertinente en el proceso. El flujo de información de cada proyecto se determina en el proceso de mapeo de la lista de actividades propuestas por el estándar y el modelo seleccionado (SPLCM)

Se debe tener en cuenta que el estándar determina que existen actividades que son requeridas (Se deben ejecutar), dentro de la sección del manejo de proyectos así como algunas que pertenecen al grupo de actividades de la sección de Soporte. Sin embargo en las otras secciones, y dependiendo del modelo (SPLCM) seleccionado, existe ciertas actividades que pueden llegar a ser requeridas.

6.6.6. Elementos del proceso del ciclo de vida del proyecto de software

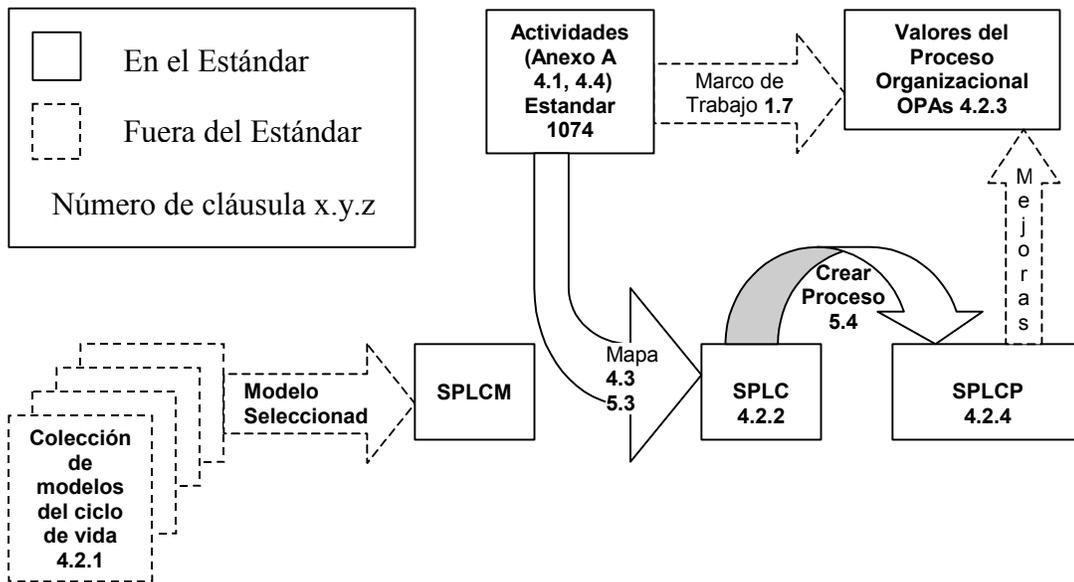


Figura 6. Elementos del proceso del ciclo de vida del proyecto de software⁸

6.6.7. Mapeo de actividades

Tal como se describió en la definición del mapeo, se puede incluir las actividades dentro de un SPLCM como:

- **Instancia.** Una actividad es mapeada como una instancia si toma toda la información de entrada, la procesa y genera toda la información de salida que se definió. Una actividad mapeada de esta forma se incluye una sola vez y se ve como un evento en el SPLC.
- **Iteración.** Una actividad se mapea como una iteración si parte (no necesariamente toda) de la información de entrada se recibe y procesa para generar parte de la información de salida. Mientras no se procese toda

⁸ IEEE, Computer Society. IEEE Standard for developing a software project life cycle process. 1074-2006. 2006.

la información de entrada y no se genere toda la información de salida se puede seguir invocando la actividad.

- **Invocación.** Esta forma de mapear una actividad permite que para una actividad que se este realizando, si se tiene alguna información de entrada que se pueda procesar por la actividad a invocar, se ejecuta la actividad, posiblemente en paralelo, se genera la información de salida posible y se retorna a la actividad que invoco la actividad.

6.6.8. Información de entrada e información de salida

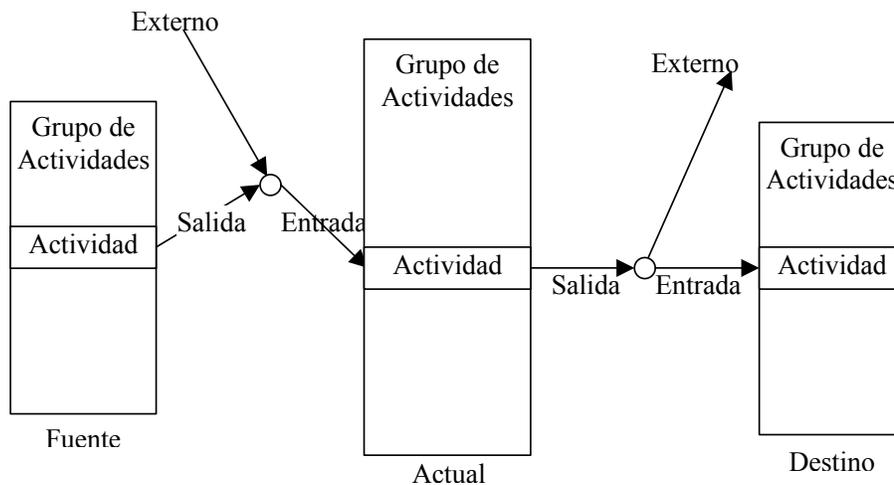


Figura 7. Flujo de Información en las actividades⁹

Mediante el mapeo de actividades y la inclusión de ellas dentro del ciclo (SPLC) se puede determinar el flujo de información ya que se cuenta con las entradas y salidas de información de las actividades.

6.6.9. Información externa

Por definición, las fuentes y destinos de la información externas están por fuera del estándar; pueden o no existir para el flujo de información, sin embargo

⁹ IEEE, Computer Society. IEEE Standard for developing a software project life cycle process. 1074-2006. 2006.

solamente en el caso en el que estén identificados se “deben” utilizar para poder dar por terminada la actividad.

6.6.10. Implementación del estándar

La implementación del estándar se lleva a cabo mediante 5 pasos, y es responsabilidad del arquitecto del proceso crear y mantener el Proceso del Ciclo de Vida del Proyecto de Software (SPLCP). Dicha actividad es llamada *Desarrollo del SPLCP*. Los pasos se describen a continuación:

- Determinar los requerimientos para el Proceso del Ciclo de Vida del Proyecto de Software.
- Seleccionar el Modelos del Ciclo de Vida del Proyecto de Software (SPLCM).
- Desarrollar el Ciclo de Vida del Proyecto de Software.
- Establecer el Proceso del Ciclo de Vida del Proyecto de Software (SPLCP).
- Validar el Proceso del Ciclo de Vida del Proyecto de Software (SPLCP).

6.6.10.1. Determinar los requerimientos para el Proceso del Ciclo de Vida del Proyecto de Software

El arquitecto del proceso debe garantizar que todos los requerimientos para el SPLCP se hayan recolectado. Los Stakeholder más importantes son identificados y sus requerimientos y expectativas transformados en un grupo de requerimientos, de tal forma que ellos, sean aceptados por los Stakeholder.

6.6.10.2. Seleccionar el Modelos del Ciclo de Vida del Proyecto de Software (SPLCM)

El arquitecto del proceso tiene la responsabilidad de identificar cual es el modelo (SPLCM), de la lista de modelos disponibles, que mas se acerque a las necesidades del proyecto.

Los pasos que se deben realizar para evaluar un modelo determinado son:

- Identificar todos los modelos (SPLCM) disponibles.
- Identificar los atributos que aplican para el sistema final deseado y para el entorno del proyecto.
- Identificar cualquier restricción que pueda afectar la selección del modelo.
- Evaluar los diferentes modelos SPLCMs, teniendo en cuenta las experiencias pasadas y las posibilidades de la organización.
- Seleccionar el Modelo que mejor satisfaga los atributos del proyecto y las restricciones.

6.6.10.3. Desarrollar el Ciclo de Vida del Proyecto de Software

Este paso consiste en el mapeo de las actividades, listadas en el estándar, frente al modelo seleccionado en el paso anterior. Las actividades que son identificadas como requeridas, se deben incluir, ya que en caso contrario se considera que el SPLC no esta en concordancia con el estándar.

Los pasos para el mapeo de actividades son:

- Colocar las actividades en una secuencia ejecutable. El orden en el cual las actividades serán ejecutadas es determinado por cuatro consideraciones:
 - El modelo SPLCM seleccionado determinara un orden inicial de actividades.

- Las restricciones en la agenda del proyecto puede hacer necesario que se reorganicen las actividades y además que sea necesario ejecutar actividades en paralelo.
- Las actividades y orden de las mismas se puede ver afectados por actividades asociadas y sus entradas y salidas.
- Las actividades que requieran información de entrada de fuentes desconocidas, se deben considerar como fuentes externas, de igual forma, si el destino de una información de salida no se conoce se debe indicar que es un destino desconocido.
- Identificar y justificar la lista de actividades no incluidas.
- Verificar el mapeo de actividades. El arquitecto del proceso debe verificar que las actividades que se incluyeron, cubran las necesidades del proyecto.

6.6.10.4. Establecer el proceso del ciclo de vida del proyecto de software (SPLCP)

El paso anterior se definió el SPLC, ahora en este paso se incluyen los OPAs para obtener el Proceso de Ciclo de Vida del Proyecto de Software.

6.6.10.5. Validar el proceso del ciclo de vida del proyecto de software (SPLCP)

El Proceso obtenido hasta este momento debe ser validado frente a los requerimientos establecidos en el primer paso.

7. DISEÑO DE LA GUIA METODOLOGICA

La guía propuesta tiene como objetivo principal introducir en los proyectos de desarrollo, actividades de administración y control que buscan asegurar la terminación del proyecto y la calidad de los productos y procesos durante el desarrollo.

La guía permite contar con un plan de trabajo que disminuya el nivel de incertidumbre que se presenta en el inicio de un proyecto, conociendo las tareas a realizar, el orden de las mismas, las personas que la realizan y los tiempos en los cuales deben ejecutarse. Adicionalmente el trabajo metodológico permite prever posibles riesgos que pueda sufrir el proyecto proponiendo alternativas de solución.

La guía metodológica tiene cuatro elementos:

- Modelo del Ciclo de vida
- Fases y actividades del ciclo de vida
- Entregables como resultado de las actividades
- Formatos de soporte

7.1. Modelo del ciclo de vida

Comenzando con la identificación del Ciclo de Vida, se ha diseñado un modelo con base en los modelos de ciclo de vida lineal y en espiral, buscando guiar de manera simple un proyecto de desarrollo de software, desde su planeación hasta su instalación.

El ciclo, se ha representado en dos gráficos para ilustrar las actividades de administración y control y las actividades específicas propias de desarrollo.

Estos ciclo trabajan de manera conjunta, y en hechos se hace énfasis en las etapas iniciales donde se realizan las actividades de planeación, requerimientos y diseño; esto con el fin de conseguir un producto que logre la aceptación del usuario y mediante las actividades de evaluación y control se busca que se construya de manera controlada en cada una de sus fases, se evalúe y se retroalimente todo el ciclo dejando registro histórico en la organización.

La etapa de construcción se hará en iteraciones sucesivas, permitiendo entregar subsistemas que se probaran por parte del desarrollador mediante un conjunto de pruebas. Este a su vez, documentará cada sistema y entregará el producto a un gestor de calidad que evaluará el producto según los indicadores establecidos por las directrices de TI para el proyecto.

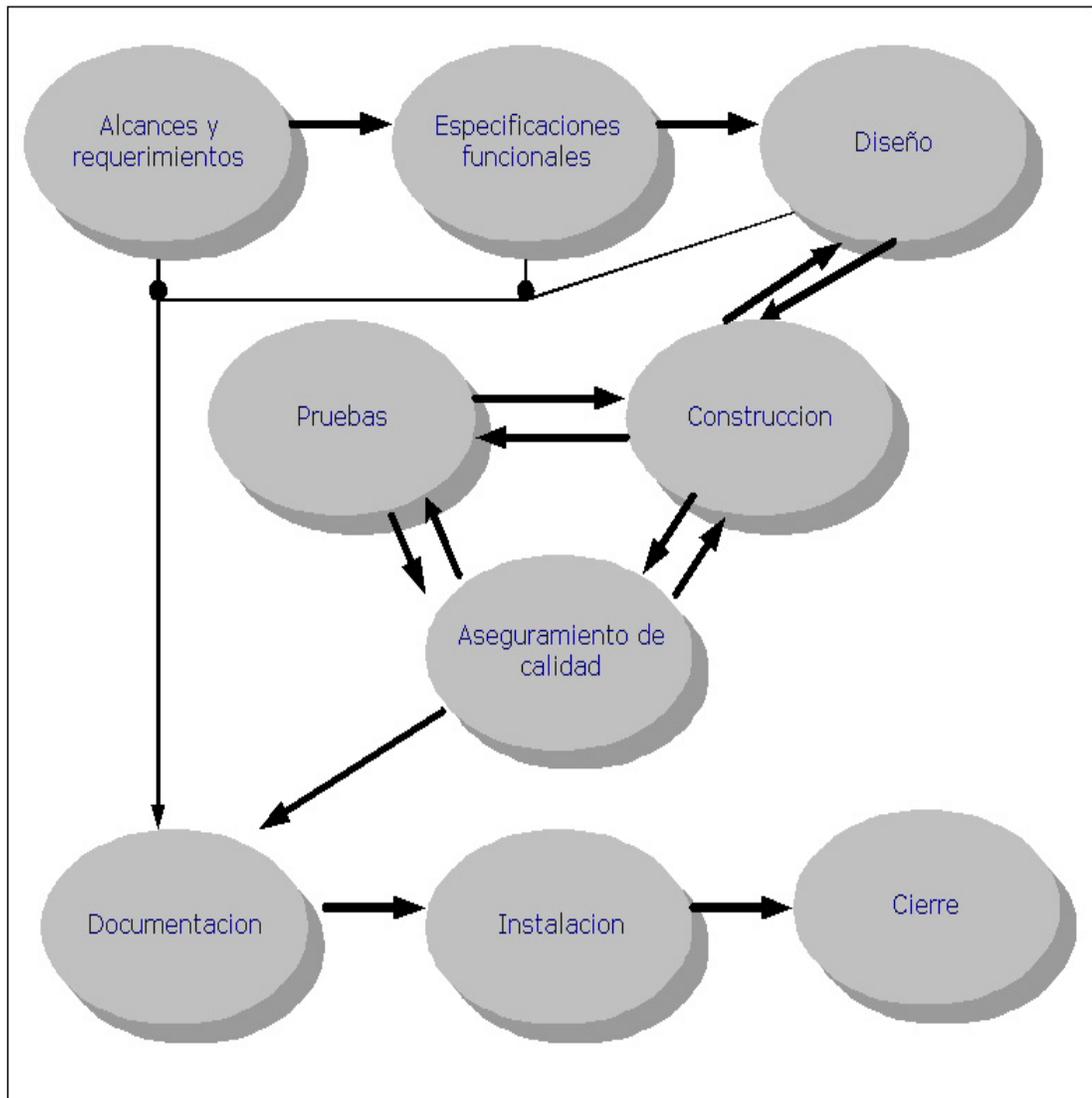


Figura 8. Ciclo Propuesto Para el Desarrollo de un Proyecto de Software

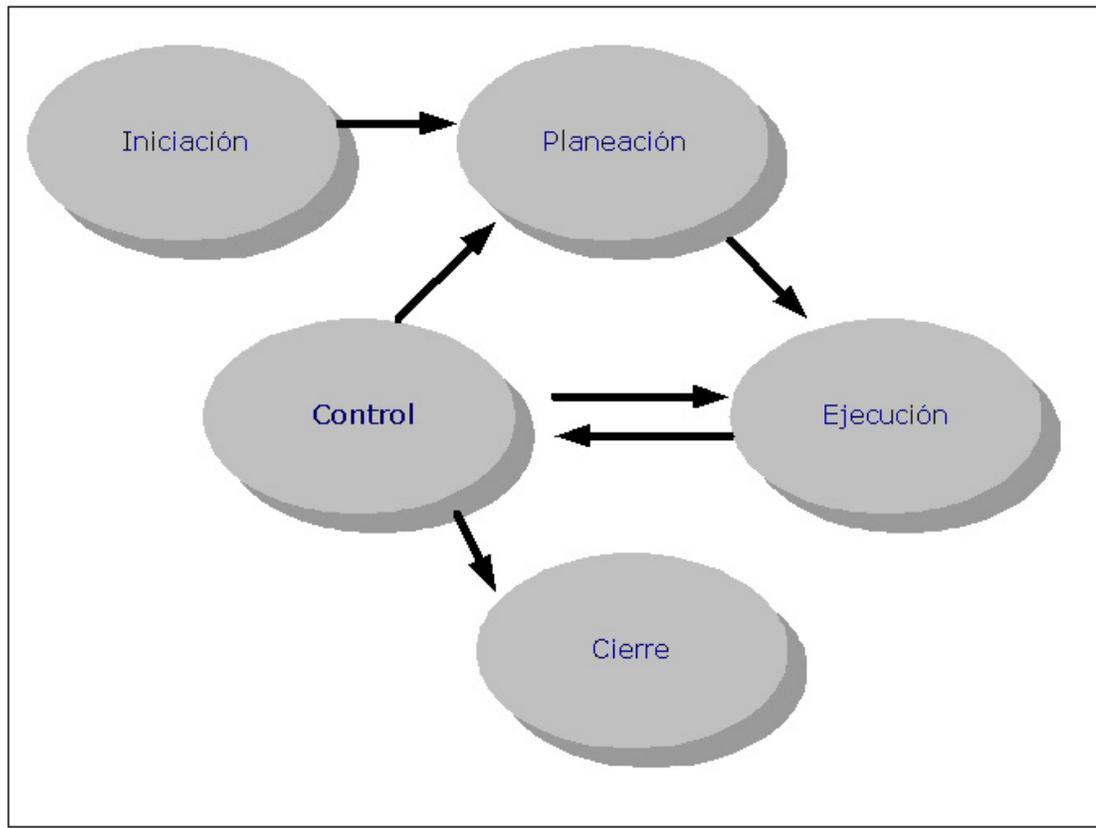


Figura 9. Ciclo Administrativo Propuesto para un Proyecto de Software

7.2. Fases del ciclo de vida para proyectos de software

La unificación de los dos ciclos, permite agrupar en cinco fases el ciclo de vida de un proyecto de software:

7.2.1. Primera fase: *Iniciación*

En esta etapa, el grupo de planeación del proyecto se dedica a explorar y conocer las necesidades o problemas de la organización recopilando datos mediante entrevistas con los usuarios y clientes, conociendo los procesos de las áreas involucradas, los informes y estadísticas que llevan, reportes históricos de los problemas presentados, actividades que se llevan manualmente, los manuales de procesos y manuales de funciones del personal que participan en el proceso.

Paralelamente, el grupo debe investigar sobre soluciones existentes en el mercado y sobre las soluciones implementadas por la competencia.

Con esta información, el grupo está en capacidad de emitir su concepto sobre el sistema requerido, dejando claramente definido los objetivos, el alcance y las estimaciones de recursos y tiempos del proyecto.

La propuesta debe sugerir alternativas de solución, en la medida de lo posible, de diferentes gamas presupuestales que se encuentren en el rango de presupuesto autorizado para el proyecto. La guía recomienda que cuando el proyecto es muy ambicioso o complejo se haga por etapas, teniendo en cuenta que la planeación debe incluir desde el inicio todas las etapas y se deje establecido la forma de ejecución con su debida aprobación.

Los formatos que están disponibles para esta fase son:

- Formato de registro de solicitudes
- Formato Lista de chequeo para inicio
- Formato Definición Conceptual
- Propuesta

7.2.2. Segunda fase: Planeación

Una vez aprobada la propuesta, la administración del proyecto y el equipo de trabajo se encarga de modelar el producto con todas sus especificaciones para establecer todos los requerimientos exigidos para su desarrollo y el plan de trabajo a realizar durante el proyecto. Es en esta fase donde se define el equipo de trabajo, los roles, funciones y responsabilidades de cada uno de los integrantes. En un proyecto típico de desarrollo de software se deben definir las funciones para

Arquitecto, Analista, Administrador de Base de datos, programadores, probadores, líder del proyecto y el gerente del proyecto. Es posible que funciones diferentes puedan ser desempeñadas por una sola persona.

En esta fase el arquitecto junto al equipo de desarrollo, modela la aplicación generando los requerimientos funcionales, las necesidades de bases de datos y la definición de interfaces, entregando la especificación del producto.

Por su parte el gerente del proyecto junto al líder define sus estrategias de administración y control, dentro de las cuales se incluye la elaboración del plan, el cronograma de trabajo, identificación de los factores de riesgo del proyecto, el plan de evaluación y los puntos de auditoría y control.

7.2.3. Tercera fase: Ingeniería

En la fase de ingeniería, se generan entregables de la aplicación o subsistemas incluidas dentro de versiones que se producen en cada una de las iteraciones de la fase con el fin de hacer entregas parciales las cuales deben estar probadas en su totalidad y aceptadas por los usuarios, garantizando que al finalizar las iteraciones el producto cubra todas las necesidades y requerimientos definidos en la fase anterior.

Es de vital importancia que durante esta fase se produzca toda la documentación pertinente tal como la documentación del código fuente, el manual técnico, el manual de usuario, y el manual de operación del sistema, solo de esta manera se garantiza que futuras modificaciones no requieran de grandes inversiones para el entendimiento de lo que ya está hecho. Igualmente se deben generar los registros de todas las evaluaciones y avances realizadas al sistema que serán la base de indicadores de ejecución del proyecto.

7.2.4. Cuarta fase: Calidad y control

En esta fase, que esta cubriendo la ejecución total del proyecto, se persigue garantizar que el producto resultante este acorde a las necesidades, que los tiempos reales de ejecución estén en concordancia con la planeación mediante la revisión de los hitos reinventando el plan de trabajo cuantas veces sea necesario con el fin de alcanzar el objetivo.

7.2.5. Quinta fase: Cierre

Por lo general el cierre del proyecto no es considerado como algo importante, sin embargo este grupo de actividades le garantiza al gerente del proyecto que el sistema entrega los resultados esperados, los cuales han sido validados y aceptados por los usuarios. De igual forma es en esta fase que se liberan los recursos que colaboraron durante la ejecución del proyecto.

7.3. Actividades y entregables de las fases del ciclo de vida

Fase/actividad	Entregable
1. Iniciación	
1.0	Proceso del ciclo de vida del proyecto de software
1.1 Definición conceptual, objetivo y alcance	
1.2 Definición del recurso del proyecto	
1.3 Elaboración de la propuesta	Propuesta
2. Planeación	
2.1 Definición del producto	Especificaciones del producto-Plan

2.2 Definición de requerimientos	Plan del proyecto
2.3 Elaboración de presupuesto (Personal requerido, Dotación Requerimientos de capacitación E Infraestructura)	Presupuesto-Plan del proyecto
2.4 Elaboración del plan de trabajo	Cronograma-Plan del proyecto
2.5 Elaboración del plan de documentación	Plan de documentación Plan del proyecto
2.6 Especificaciones funcionales	Especificaciones del producto
2.7 Arquitectura de la solución	Especificaciones del producto
2.8 Requerimientos de base de datos	Especificaciones del producto
2.9 Diseño de interfases	Especificaciones del producto
Ingeniería	
3.1 Construcción	Código fuente, componentes, base de datos
3.2 Pruebas funcionales	Indicadores, reportes
3.3 Documentación	Manuales, actas, informes
3.4 Instalación del software	Software instalado
Control y calidad	
4.1 Aseguramiento de calidad	Registros históricos del proyecto
4.2 Configuración del sistema	Software configurado
Cierre	
5.1 Validación de resultados	Reportes de control
5.2 Aceptación del proyecto	Acta de entrega

Tabla 2. Fases, actividades y entregables de la guía

7.4. Actividades de las fases del ciclo de vida

7.4.1. Fase de Iniciación

7.4.1.1. Definición conceptual, objetivo y alcance

De acuerdo a las entrevistas a usuarios, registros históricos relativos a problemas y a actividades manuales relacionadas con los procesos involucrados en el proyecto de desarrollo de software, la criticidad operacional y la sensibilidad de los datos se determina las ideas o necesidades para el nuevo sistema y se obtiene la información para la elaboración preliminar de las necesidades funcionales y no funcionales del proyecto.

También se debe incluir el plan que describa los productos, los requerimientos de seguridad organizacional, los planes de trabajo y las restricciones de recursos que se deben tener en cuenta para la ejecución del proyecto.

Insumos

- Registros históricos
- Entrevistas a usuarios
- Manuales de funciones
- Manuales de operaciones
- Flujos de procesos
- Estimaciones
- Políticas de seguridad de la organización
- Información de riesgos reportados
- Solicitud de clientes
- Problemas reportados
- Restricciones de la organización

Salidas

- Declaración preliminar de requerimientos (Ver Formato Documentación)
- Registros históricos del proyecto
- Plan de riesgos
- Propuesta del proyecto (Ver formato Propuesta del Proyecto)
- Definición conceptual (Ver formato Definición Conceptual)
- Plan de documentación

7.4.1.2. Definición de recursos del proyecto

Se determinan las necesidades de personal, presupuesto, equipo, computadores, infraestructura física e imprevistos del proyecto.

Insumos

- Registros históricos
- Estimaciones
- Recursos
- Requerimientos de software del sistema funcional
- Proceso del ciclo de vida

Salidas

- Plan de administración del proyecto

7.4.1.3. Elaboración de la propuesta

Consiste en la elaboración de un documento que describa los objetivos del proyecto, el alcance, la visión de la solución, presente una estimación de costos en términos de tiempo, inversión y recursos. (Ver Formato Propuesta Proyecto).

Insumos:

- Definición conceptual, objetivo y alcance
- Definición de recursos del proyecto

Salidas:

- Propuesta del Proyecto (Ver Formato Propuesta Proyecto)

7.4.2. Fase de planeación

7.4.2.1. Definición del producto

Esta actividad requiere del conocimiento profundo de las necesidades para poder establecer en detalle el producto.

Teniendo en cuenta la información de los recursos, el presupuesto disponibles, se preparan propuestas que permitan tomar la decisión del desarrollo del sistema, incluyendo en ellas los beneficios y restricciones; recomendando la propuesta que se ajuste mas a las necesidades.

Si es necesario debe documentarse con aspectos legales y estándares existentes.

Insumos:

- Declaración preliminar de requerimientos
- Solicitudes de usuarios
- Definición conceptual
- Propuestas potenciales
- Recomendaciones

Salidas:

- Especificación del producto

7.4.2.2. Definición de requerimientos

Con base en las especificaciones del producto, se elabora una declaración preliminar de requerimientos del producto para establecer posteriormente las necesidades de software, recursos humanos e interfaces.

Insumos:

- Especificación del producto

Salida

- Declaración preliminar de requerimientos

7.4.2.3. Elaboración del presupuesto

Ajusta del presupuesto presentado en la propuesta del proyecto, de acuerdo a las especificaciones del producto y a la declaración de requerimientos.

Se debe incluir en el presupuesto, personal requerido, dotación, requerimientos de capacitación, infraestructura y viáticos.

Insumos

- Especificaciones del producto
- Requerimientos
- Propuesta del proyecto

7.4.2.4. Elaboración del plan de trabajo

El plan recopila de forma detallada la información para la administración del proyecto, generando el detalle de la organización del mismo y la asignación de responsabilidades. En esta actividad se identifican y describen las tareas a

desarrollar durante la ejecución del proyecto, identificando las actividades que permitirán evaluar y controlar los tiempos y logros durante el avance del proyecto. Para tal fin se debe incluir en el plan actividades de auditoria, revisión y pruebas definiendo los criterios de aceptación o rechazo. Igualmente deben quedar identificados los métodos de evaluación los propósitos y el alcance. Igualmente el plan debe incluir actividades encaminadas al manejo de la configuración del sistema, ofreciendo herramientas, procedimientos, técnicas y metodologías.

Es necesario que las actividades estén priorizadas, con el fin de mantener el orden lógico en la ejecución de las tareas que no ocasionen retrasos al proyecto.

El plan debe ser concebido de una manera flexible que permita durante las actividades de revisión y control, alinear los requerimientos funcionales y no funcionales en cada fase del proyecto. Por otra parte el plan debe incluir las actividades que permitan la entrega de la documentación de acuerdo a las necesidades de información.

Insumos:

- Métodos de recopilación y análisis de datos
- Requerimientos de software
- Lista de entregables
- Objetivos de seguridad
- Información de riesgos
- Recomendaciones
- Estimaciones
- Recursos

Salidas:

- Plan de trabajo

7.4.2.5. Elaboración del plan de documentación

En esta actividad se diseña, prepara y mantiene la documentación del proyecto, especificando el objetivo del documento, el público al que se dirige y el contenido.

Insumos:

- Formato de Documentación
- Información de la planeación del manejo del proyecto de software

Salidas:

- Plan de documentación
- Documentación
- Manuales
- Actas
- Informes de avance

7.4.2.6. Especificaciones funcionales

Identifica las funciones del sistema, los requerimientos no funcionales y restricciones conocidas, analizando el impacto con el fin de delimitarlos y usarlos para desarrollar la arquitectura del sistema.

Insumos

- Recomendaciones
- Definición conceptual

Salidas:

- Descripción funcional del sistema

7.4.2.7. Arquitectura de la solución

Esta actividad permite transformar las necesidades funcionales en planos y rutas a seguir para el desarrollo del sistema siguiendo las metodologías y directrices establecidas por la organización.

Insumos:

- Plan de trabajo
- Vulnerabilidades de seguridad conocidas
- Requerimientos del sistema
- Restricciones

Salida

- Diseño detallado

7.4.2.8. Requerimientos de base de datos

Siguiendo los requerimientos de software y sus propiedades y funciones dentro del sistema se modela el diseño conceptual, lógico y físico del sistema que permite describir las entidades de datos, sus atributos y relaciones. Adicionalmente ;a estructura de datos y su mapeo dentro de un esquema lógico dependiente de la implementación.

Insumos

- Requerimientos de software

Salida

- Diseño de la base de datos

7.4.2.9. Diseño de interfases

En esta actividad se define lo relacionado con el usuario, interfases de software y hardware de acuerdo a lo especificado en los requerimientos de software.

Insumos

- Requerimientos de software
- Especificaciones técnicas de otras aplicaciones

7.4.3. Fase de ingeniería

7.4.3.1. Construcción

Consiste en la transformación del diseño detallado en códigos programables, integrados a la base de datos. Esta actividad entrega el código fuente, el código ejecutable y drivers y exige que el software se entregue probado y debidamente documentado. El código entregado debe seguir las reglas de nombramiento, empaquetamiento y versiones.

Insumos

- Base de datos
- Configuración
- Requerimientos de software
- Diseño detallado
- Estándar de codificación del software
- Plan de documentación
- Estándar para el manejo de versiones

Salida

- Código ejecutable
- Código Fuente
- Base de datos

7.4.3.2. Pruebas funcionales

Define los pasos a seguir en las pruebas, estableciendo los ítems a probar, el tipo de prueba, los datos de prueba, los resultados esperados, el número de iteraciones y los componentes del ambiente de prueba.

Esta activa registra los resultados alcanzados permitiendo el análisis de indicadores, que evalúan los avances del proyecto, recomendando ajuste al plan de trabajo si es necesario y enriqueciendo la base de conocimientos de la organización.

Las pruebas pueden generar cambios en la construcción del sistema,

Insumos

- Plan de evaluación
- Requerimientos de software
- Diseño detallado

Salida

- Software probado
- Registros de evaluación
- Recomendaciones y ajustes

7.4.3.3. Documentación

Esta actividad produce la documentación operación desde el diseño detallado, requerimientos del sistema, manual de usuario, manual técnico y demás documentación establecida en el plan de documentación.

Insumos

- Plan de documentación
- Requerimientos de software
- Diseño detallado

Salida

- Documentación operacional

7.4.3.4. Instalación del software

Se procede con la instalación del software una vez se tiene seguridad del ambiente operacional y disponibilidad de los recursos requeridos. Debe entregar un reporte detallado de la instalación y los problemas encontrados. En esta actividad el software probado, la base de datos, la documentación de instalación y operacional deben empaquetarse con un solo nombre para facilitar la distribución e instalación del software.

Insumos

- Software probado y empaquetado
- Recursos requeridos
- Documentación

Salida

- Software instalado
- Reporte de instalación

7.4.4. Fase de control y calidad

7.4.4.1. Aseguramiento de calidad

Se revisa en esta actividad los puntos de mejoras que pueden ser benéficos para el proyecto de acuerdo a los resultados arrojados en las evaluaciones. Esta actividad se hace a lo largo del ciclo de vida permitiendo eliminar los defectos en los requerimientos de software, diseño preliminar, diseño detallado, código y documentación. Las revisiones pueden ser formales o informales permitiendo la alineación constante del proyecto mediante acciones correctivas o planes de contingencia.

Insumos

- Ítems para revisión
- Requerimientos de software
- Registro de control de cambios
- Reporte de problemas
- Resultados de auditoria

Salida

- Reporte administrativo del proceso
- Recomendaciones de mejoras del proceso
- Resultados de evaluación de requerimientos de software, diseño, código y documentación

7.4.4.2. Configuración del sistema

De acuerdo a los parámetros de operación establecidos, se busca en esta actividad que el sistema quede en las condiciones iniciales con las cuales se hará la puesta en marcha de la solución.

Insumos

- Parámetros del sistema
- Documentación operacional

Salida

- Sistema configurado

7.4.5. Fase de Cierre

7.4.5.1. Validación de resultados

En esta actividad se hace una lista de chequeo de los resultados esperados de la operación del sistema en tiempo real, consideraciones de seguridad, usabilidad y pruebas.

Se hace un balance en termino de la ejecución presupuestal, justificando las variaciones presentadas en termino de dinero y tiempo y cada ajuste o alineación del plan.

Se listan las posibles mejoras que se le puede hacer al sistema con su justificación.

Insumos

- Base para evaluación
- Información de reportes de error
- Información reportada de posdesarrollo
- Recomendaciones de mejoras del proceso
- Información de resultados de auditoria
- Información reportada de las pruebas funcionales

Salida

- Información de evaluación

7.4.5.2. Aceptación del proyecto

Esta actividad corresponde a el cierre del proyecto generando, recopilando, verificando y distribuyendo toda la información pertinente para concluir el proyecto formalmente.

Adicionalmente garantiza la entrega formal del proyecto a los responsables de la operación del sistema y deja constancia de aceptación del proyecto de las partes interesadas.

Insumos

- Aceptación del cliente
- Necesidades de mejoras
- Registro histórico del proyecto
- Información de administración de proyecto reportada
- Información de problemas reportada

Salida

- Acta de garantía de aceptación

8. CONCLUSIONES

El profesionalismo en todas las disciplinas del saber está cobrando mucha importancia y cada vez son mayores los esfuerzos de las personas y organizaciones en el estudio y aplicación de estándares y herramientas que le den valor agregado y calidad a todos sus procesos.

La globalización de la economía y los avances tecnológicos exige un alto grado de competitividad, por esta razón las áreas de tecnología deben incorporar técnicas como atributo de calidad, administración y control en el desarrollo de software.

La presente guía metodológica pretende ser una herramienta que de forma sencilla y concreta le permita a las organizaciones medianas y pequeñas comenzar a organizar su área de desarrollo de software sin generar traumatismos durante su implementación en los procesos del día a día.

Para la guía se tuvo en cuenta el esquema propuesto por el estándar IEEE 1074-2006, sin desconocer la existencia de otros.

La aplicación de la guía no es rigurosa en el cumplimiento de las actividades ni estática; así quienes la usen pueden ajustarla a sus necesidades y de manera progresiva ir incrementando las actividades y ofreciendo nuevas características, insumos y salidas del tal manera que se dinamice según las exigencias y los recursos.

Teniendo en cuenta lo anterior, se puede concluir:

1. La carencia de una metodología que asista los desarrollos de software contribuye en un alto porcentaje a la problemática actual que viven las áreas de

tecnología. Así que independiente del estándar a seguir, el área de desarrollo debe hacer un esfuerzo para replantear sus procesos, tener claramente identificados roles y responsabilidades, priorizar sus problemas e iniciar como un proyecto con alta prioridad la implementación de actividades y formatos que se ajusten a sus necesidades.

2. El uso de una metodología permite que personal novato se incorpore fácilmente a los proyectos.

4. El registro histórico de datos, evaluaciones permanentes y elaboración de la documentación a lo largo del proceso permite tener información histórica de los proyectos que pueden reutilizarse en otros proyectos similares

5. El esfuerzo inicial para la implementación de una metodología para proyectos de software se ve recompensando en aspectos cualitativos como calidad, profesionalismo, certeza en la ejecución de tareas, conocimiento y dominio de temas por parte del personal del área y aspectos cuantitativos expresados en disminución de sobrecostos por esfuerzo en mantenimiento, re-procesos, nuevos tiempos de entrega, etc.

8.1. Aportes

La presente Guía Metodológica provee una herramienta sencilla y fácil de usar para las compañías que quieran organizar su proceso de desarrollo de software, lo que se ha hecho es organizar una cantidad de información metodológica y definir la lista de actividades con sus insumos y salidas las cuales se pueden implementar fácilmente durante el desarrollo.

Pero algo que marca la diferencia frente otras metodologías, es el compendio de formatos, suministrados, fáciles de utilizar y que recogen la mayor cantidad de

información que debe ser útil para el proyecto, dichos formatos van desde la definición conceptual del proyecto, pasan por listas de control de actividades y así sucesivamente, incluidos para brindarle un soporte a los integrantes del equipo de trabajo de los proyectos de software.

8.2. Trabajos Futuros

Futuras ampliaciones de la guía pueden estar encaminadas a:

- Considerar aspectos de otros estándares para proyectos de desarrollo de software que permitan incorporar nuevas posibilidades y herramientas de trabajo
- Incluir estándar de programación, que brinde un esquema de trabajo en la construcción del producto.
- Aplicar la guía metodología en una de las empresas que participaron en el diagnóstico empresarial con el fin de poder evaluar la efectividad de la misma y retroalimentar la guía con nuevas actividades y formatos.
- Profundizar en la definición de las actividades para generar mayor grado de comprensión en la ejecución.

9. BIBLIOGRAFÍA Y REFERENCIAS

ISO/IEC 9126-1: 2001 (E), International Standard “*Software Engineering - Product Quality- Part 1: Quality model*”.

FERRE, Xavier; JURISTO, Natalia: "How to integrate usability into the software development process," *icse*, pp. 1079-1080, Proceeding of the 28th international conference on Software engineering (ICSE '06), 2006

IEEE, Computer Society. IEEE Standard for developing a software project life cycle process. 1074-2006. 2006. New Cork, 28 de julio de 2006.

LETELIER, Patricio. Introducción a Rational Unified Process (RUP). Disponible en www.dsic.upv.es/~letelier/pub; Departamento Sistemas Informáticos y Computación (DSIC). Universidad Politécnica de Valencia (UPV) - España

LOPEZ, Carmelo. Revista de ingeniería informática del CIIRM. Disponible en internet en: <http://www.cii-murcia.es>

PRESSMAN, Roger. *Software Ingeneering, a Practitioner’s Approach*. Sexta Edición. McGraw-Hill Profesional. 2005.

PROJECT Management Institute, Project Management body of knowledge, PMBOOK Guide, Newtown Square, Pennsylvania USA, Edition 2000

ANEXOS

- A. Encuesta para el diagnóstico de metodología de desarrollo de software
- B. Mapeo de Actividades del estándar IEEE 1074-1006 en las fases del Ciclo de Vida
- C. Lista de actividades del estándar IEEE 1074-1006 frente a las actividades pertenecientes a las fases del ciclo de vida
- D. Formatos

Anexo A

ENCUESTA PARA EL DIAGNOSTICO DE METODOLOGIA DE DESARROLLO DE SOFTWARE

Objetivo: Conocer la situación al interior del Área de Tecnología de las organizaciones, referente a la aplicación de metodologías en la gestión de proyectos de desarrollo de software

Dirigido a: Directores de áreas de TI o directores de proyectos de desarrollo de software

1. ¿Cuenta su empresa con un área de Tecnología?

Si No

Si su respuesta es afirmativa, por favor continúe con la encuesta.
Si su respuesta es negativa, Gracias por su tiempo.

2. ¿Tiene el área de TI, de su organización, definidas la misión y visión del área?

Si No

3. ¿Están la misión y visión del área alineadas con la estrategia de la organización?

Si No

4. ¿El área de TI define planes estratégicos anuales?

Si No

5. ¿Tiene el área de TI definidas metas y recursos para proyectos de desarrollo de software (nuevos sistemas, mantenimiento o desarrollo de integración de aplicaciones)?

Si No

6. Dentro de la estructura funcional del área de TI, ¿Se cuenta con un área de desarrollo de software?

Si No Si su respuesta es afirmativa pase a la pregunta 7, en caso contrario a la pregunta 8.

7. ¿Cuántas personas laboran en el área de desarrollo? _____

8. ¿En el área de TI se cuenta con personas dedicadas a la gerencia de proyectos?

Si No

9. ¿El área tiene una metodología para el proceso del ciclo de vida de proyectos de software?

Si No

10. ¿En el área se aplican estándar de desarrollo?

Si No

En caso de respuesta afirmativa, ¿Cuales?

11. ¿Tienen control de fuentes y versiones?

Si No

12. ¿Qué porcentaje del tiempo invierten en mantenimiento de software?

13. ¿Tienen definido un procedimiento y los formatos respectivos para el levantamiento de requerimientos?

Si No

14. ¿Documentan los desarrollos de software?

Si No

15. ¿Documentan los mantenimientos realizadas a los sistemas de información?

Si No

16. ¿Cual es la mayor problemática que tiene el área de desarrollo?

17. ¿Considera importante implementar una metodología para gestionar los procesos del ciclo de vida de los proyectos de desarrollo de software?

Si No

¿En caso de respuesta de afirmativa, explique el porque?

Anexo B

Mapeo de Actividades del estándar IEEE 1074-1006 en las fases del Ciclo de Vida

Actividades del estándar	Fases del ciclo				
	Iniciación	Planeación	Ingeniería	Control y calidad	Cierre
1.1 Desarrollar el proceso de ciclo de vida de proyecto de software	x				
1.2 Estimaciones	x				
1.3 Determinar los recursos del proyecto	x				
1.4 Indicadores	x				
1.5 Determinar objetivos de seguridad	x				
1.6 Plan de evaluaciones		x			
1.7 Plan de administración de configuración		x			
1.8 Plan de documentación	x	x			
1.9 Plan de administración del proyecto			x		
1.10 Administración de riesgos	x	x	x		
1.11 Identificar necesidades de mejoras del proceso del ciclo de vida del proyecto de software				x	x
1.12 Guardar registros			x		
1.13 Recopilación y análisis de indicadores			x		
1.14 Cierre del proyecto					x
2.1 Definición conceptual	x	x			
2.2 Declaración preliminar de requerimientos		x			
2.3 Estudio de viabilidad		x			
2.4 Finalizar la idea o necesidad		x			
2.5 Especificaciones funcionales		x			

2.6. Arquitectura del sistema		x			
2.7 Requerimientos del sistema		x			
3.1 Definir y desarrollar los requerimientos		x			
3.2 Requerimientos de interfases		x			
3.3 Priorizar e integrar los requerimientos		x			
3.4 Diseño de arquitectura		x			
3.5 Diseño de la base de datos		x			
3.6 Diseño de interfases		x			
3.7 Diseño detallado		x			
3.8 Creación de código ejecutable			x		
3.9 Creación de documentación operacional			x		
3.10 Integración			x		
3.11 Administración de versiones de software		x	x		
4.1 Distribución de software			x	x	x
4.2 Instalación de software			x	x	
4.3 Aceptación del software en ambiente operacional				x	
5.1 Revisión de comportamiento	x	x	x	x	X
5.2 Procedimientos de pruebas de desarrollo			x		
5.3 Pruebas de ejecución			x		
5.4 Resultados de evaluación	x		x		X
5.5 Implementación de documentación		x			
5.6 Producir y distribuir documentación		x			

Anexo C

Lista de actividades del estándar IEEE 1074-1006 frente a las actividades pertenecientes a las fases del ciclo de vida

Fase/actividad	Actividades del estándar
1. Iniciación	
1.0	1.1 Desarrollar el proceso de ciclo de vida de proyecto de software
1.1 Definición conceptual, objetivo y alcance	1.2 Estimaciones 1.3 Indicadores 1.4 Determinar objetivos de seguridad 1.8 Plan de documentación 1.10 Administración de riesgos 2.1 Definición conceptual 5.1 Revisión de comportamiento
1.2 Definición del equipo propuesto del proyecto	1.3 Determinar los recursos del proyecto
1.3 Aprobación de la propuesta	
2. Plantación	
2.1 Definición del producto	2.1 Definición conceptual 2.3 Estudio de viabilidad 2.4 Finalizar la idea o necesidad
2.2 Definición de requerimientos	2.2 Declaración preliminar de requerimientos
2.3 Elaboración de presupuesto (Personal requerido, Dotación Requerimientos de capacitación E Infraestructura)	1.3 Determinar los recursos del proyecto
2.4 Elaboración del plan de trabajo	1.6 Plan de evaluaciones 1.7 Plan de administración de configuración 1.8 Plan de documentación 1.9 Plan de administración del proyecto 1.10 Administración de riesgos 3.3 Priorizar e integrar los requerimientos Revisión de comportamiento 5.6 Producir y distribuir documentación

2.5 Elaboración de mecanismos de reporte	5.5 Implementación de documentación
2.6 Determinación de factores que afectan el proyecto	5.1 Revisión de comportamiento
2.7 Establecimientos de puntos de control y entregas	5.1 Revisión de comportamiento
2.8 Especificaciones funcionales	2.5 Especificaciones funcionales 2.7 Requerimientos del sistema
2.9 Arquitectura de la solución	2.6 Arquitectura del sistema 2.7 Requerimientos del sistema 3.4 Diseño de arquitectura 3.7 Diseño detallado
2.10 Requerimientos de base de datos	2.7 Requerimientos del sistema 3.5 Diseño de la base de datos
2.11 Requerimientos de internases	2.7 Requerimientos del sistema 3.6 Diseño de interfases
Ingeniería	
3.1 Construcción	3.8 Creación de código ejecutable 3.10 Integración 3.11 Administración de versiones de software 4.1 Distribución de software
3.2 Pruebas funcionales	1.12 Guardar registros 1.13 Recopilación y análisis de indicadores 3.10 Integración Revisión de comportamiento Procedimientos de pruebas de desarrollo Pruebas de ejecución
3.3 Documentación	3.9 Creación de documentación operacional Resultados de evaluación
3.4 Instalación	4.2 Instalación de software
Control y calidad	
4.1 Aseguramiento de calidad	1.11 Identificar necesidades de mejoras del proceso del ciclo de vida del proyecto de software 4.1 Distribución de software

	5.1 Revisión de comportamiento
4.2 Configuración del sistema	4.2 Instalación de software
Cierre	
5.1 Validación de resultados	1.12 Identificar necesidades de mejoras del proceso del ciclo de vida del proyecto de software 5.1 Revisión de comportamiento 5.4 Resultados de evaluación
5.2 Aceptación del proyecto	1.14 Cierre del proyecto 4.1 Distribución de software 4.3 Aceptación del software en ambiente operacional

Anexo D

Formatos de la guía metodológica.

Los siguiente son los formatos que se incluyen en la guía con el propósito que sean utilizados en los proyectos de desarrollo de software.

- 1. Certificado de aceptación**
- 2. Formato Definición Conceptual**
- 3. Formato Definir Set de Pruebas**
- 4. Formato Documentación**
- 5. Formato Especificaciones Funcionales**
- 6. Formato Lista chequeo para inicio**
- 7. Formato Modificación Base Datos**
- 8. Formato Propuesta Proyecto**
- 9. Formato Registro de Solicitudes**
- 10. Formato Reporte Ejecución Actividades**
- 11. Formato Solicitud mejora, cambio, nuevo desarrollo, error**
- 12. Formato Solicitud Paso a Producción**

Formato: Certificado de Aceptación

Nombre Empresa:	Certificado de Aceptación
Fecha de Elaboración: (AAA/MM/DD)	Fecha de Aprobación: (AAA/MM/DD)
Elaborado por:	Aprobado por:
Area Funcional	Proyecto/Tarea/Desarrollo:
Tipo de Aceptación: (Propuesta, Plan, Especificación, Documentación)	Descripción:

La documentación entregada y aceptada mediante el presente certificado, ha sido revisada por _____ de tal manera que entiende claramente los temas expresados en el documento y se da por aceptado el contenido, de acuerdo a los criterios de evaluación previstos para tal fin.

Cualquier problema ocasionado durante los siguientes pasos del proceso, debe ser notificado por escrito, durante los tres días siguientes, con el fin de tener el registro y solucionar cualquier inconveniente, caso contrario se da por entendido que todo se ha aceptado a satisfacción.

Nombre:
C.C.

Nombre:
C.C.

Formato: Definición Conceptual

Nombre Empresa:	Formato para Definición Conceptual
Proyecto/Tarea/Desarrollo:	Fecha de Elaboración: (AAA/MM/DD)
Elaborado por:	
Actualizaciones del documento:	

Definición Conceptual:

Temas por Resolver:

Comentarios:

Tips:

1. En la definición conceptual se debe:
 - a. Describir el problema o situación a solucionar indicando sus necesidades, sin detallar asuntos relevantes a "Sistemas" propiamente dichos.
 - b. El lenguaje utilizado no debe ser técnico, por tanto se habla de sistemas solamente en caso de ser estrictamente necesario.
 - c. Se debe escribir teniendo en cuenta que quienes van a leer el documento son personas diferentes a analistas y programadores, tales como gerentes, usuarios finales quienes requieren de un lenguaje sencillo y explícito.
 - d. En lo posible se debe limitar el alcance de la solicitud indicando los aspectos que serán tenidos en cuenta y los que se excluyen para futuras mejoras.
2. En los temas por resolver, se debe dejar constancia de todos esos detalles que por alguna razón no se han aclarado o definido, de tal forma que sean tenidos en cuenta para el desarrollo.

Formato: Definición de Sets de Prueba

Nombre Empresa:	Formato para Definición de Sets de Pruebas
Fecha de Elaboración: (AAA/MM/DD)	Fecha de Aprobación: (AAA/MM/DD)
Elaborado por:	Base de Datos
Proyecto/Tarea/Desarrollo:	

- I. Quien Debe realizar la Prueba.
- II. Programas a Probar.
- III. Procedimientos Almacenados a Probar.
- IV. Tablas a Actualizar en la Prueba
- V. Resultados Esperados en las Tablas
- VI. Resultados Esperados del Procesamiento
- VII. Descripción Detallada de la Prueba

Formato: Documentación

Nombre Empresa:	Formato para Documentación
Nombre Documento:	Tipo de Documento
Fecha de Elaboración: (AAA/MM/DD)	Fecha de Aprobación: (AAA/MM/DD)
Elaborado por:	Aprobado por:
Actualizaciones del documento:	

Tips:

Este documento debe ser diligenciado por los integrantes del proyecto que creen, actualicen y utilicen la documentación.

En la sección de actualizaciones del documento se deben indicar fechas, personas y asuntos modificados del documento.

Debe especificarse los tipos de documentos:

- Manual de usuario
- Manual técnico
- Manual del sistema

Fecha: 27/07/2007 Hora:21:12
Formato Aprobado por:
Cargo quien aprueba:

Pág. 1 de 1
Formato Documentación. [Ver 00]

Formato: Especificaciones Funcionales

Nombre Empresa:	Formato para Especificaciones Funcionales
Fecha de Elaboración: (AAA/MM/DD)	Proyecto/Tarea/Desarrollo:
Elaborado por:	Aprobado por:
Actualizaciones del documento:	

Descripción de la Necesidad Funcional.

Tips:

En la descripción de la necesidad funcional, se debe:

1. Describir como debe funcionar el sistema para dar solución a la necesidad indicada en la definición conceptual.
2. Excluir indicaciones de bases de datos, transacciones, históricos, marcas en registros, etc. Ya que son temas que se deben tener en cuenta en los siguientes puntos.
3. El documento esta orientado a usuarios finales y al grupo de desarrollo, por tanto el lenguaje utilizado NO debe ser técnico (en lo posible.).
4. Cada necesidad funcional debe ser pequeña e implementable, por tanto se pueden elaborar varias descripciones de la necesidad funcional para una misma definición conceptual.

Ejemplo: (Definición conceptual: Compra por Internet)

El usuario debe ingresar al sitio, indicando usuario y contraseña, posteriormente debe seleccionar los articulos que va a comprar generando un "carrito de compra". Cuando se haya completado el carrito debe ver la liquidación incluyendo impuestos y descuentos. Posteriormente se solicitara el número de la tarjeta de crédito para hacer efectiva la compra.

Definición de Funcionalidades a Desarrollar.

Tips:

1. Enumerar el inventario de funcionalidades a desarrollar, incluyendo cosas como actualización de registros, generación de históricos, etc.
2. En caso de cambios en programas, se debe enumerar los cambios que se van a realizar y donde.

Ejemplo: (Definición conceptual: Compra por Internet)

- Crear una pantalla para solicitar usuario y password
- Procedimiento de autenticación de usuarios

Fecha: 27/07/2007 Hora:21:11

Archivo: Formato Especificaciones Funcionales.doc

Pág. 1 de 1

Nombre Empresa:	Formato para Especificaciones Funcionales
-----------------	---

- Consulta de artículos e inventario
- Creación de carritos
- Inclusión y eliminación de artículos dentro del carrito

Especificación Técnica del desarrollo. (Debe incluir descripción programas, pantallas, reportes)

Tips:

En este punto se deben detallar cada una de las funcionalidades a desarrollar, describiendo programas, dibujando pantallas, realizando diagramas de UML.

Ejemplo: (Definición conceptual: Compra por Internet)

Procedimiento de autenticación de usuarios: Se debe crear verificar que el usuario y el password indicados por el usuario sean validos mediante la ejecución de un procedimiento almacenado que consulta la base de datos.

El procedimiento almacenado (PAAUTENTICA) debe recibir como parámetros el usuario y su contraseña y.....

Modificaciones o Definición de Nuevas Tablas en la Base de datos

Tips:

Elaborar una lista de los requerimientos de actualización de la base de datos, donde se incluyan creación o modificación de tablas, índices, restricciones etc., las cuales se deben discutir con el DBA (Administrador de la Base de Datos), para generar el formato de "Modificación de Base de Datos".

Formato: Lista de Chequeo para Inicio de Proyecto

Nombre Empresa:	Lista de Chequeo para Inicio del Proyecto
Fecha de Elaboración: (AAA/MM/DD)	Fecha de Aprobación: (AAA/MM/DD)
Elaborado por:	Aprobado por:
Proyecto/Tarea/Desarrollo	

1. Recopilación de la documentación requerida
2. Definición conceptual
3. Definición del plan de trabajo
4. Definición del equipo de trabajo
5. Definición de estándares requeridos.
6. Creación de la librería para documentación del proyecto
7. Definición de roles del equipo de trabajo
8. Definición de los requerimientos de recursos adicionales

Formato: Propuesta del Proyecto

Nombre Empresa:	Propuesta del Proyecto
Proyecto:	
Fecha de Elaboración: (AAAA/MM/DD)	Fecha de Aprobación: (AAAA/MM/DD)
Elaborado por:	Aprobado por:
Actualizaciones del documento:	

Reconocimiento.

1. Objetivos
2. Descripción del problema
3. Expectativas de los usuarios y clientes

Formulación de la Propuesta

1. Alcance
2. Limitaciones
3. Recomendaciones

Estimación

1. Costos
2. Tiempos
3. Recursos

Formato: Solicitud de Mejora, Cambio, Nuevo Desarrollo o Corrección de Error

Nombre Empresa:	Formato para Solicitud de Mejora, Cambio, Nuevo Desarrollo o Corrección de Error
Nro Solicitud:	Tipo de Solicitud:

Fecha de Elaboración: (AAA/MM/DD)	Fecha de Recepción: (AAA/MM/DD)
Solicitado por:	Aprobado por:
Area Funcional:	Prioridad:

Descripción de la solicitud:

Situación actual:

Acciones posibles:

Fecha Esperada de la Solución (AAAA/MM/DD)

Tips:

1. En tipo de solicitud se debe indicar si es una tarea, mejora, nuevo desarrollo, corrección de un mal funcionamiento (bug) o un cambio.
2. El Nro de Solicitud debe ser un consecutivo asignado en el área de desarrollo para control interno.
3. Para la descripción de la solicitud describa detalladamente lo que se requiere.
4. En situación actual indique como se esta solucionando o realizando la tarea en este momento.
5. En Acciones posibles enumere cuales son las soluciones posibles a la problemática, se puede considerar el hecho de no hacer nada.
6. En Fecha esperada, plantee una fecha en el futuro en la cual deba esta solucionada la problemática.

Costo Estimado	Tiempo Estimado
Fecha de Inicio	Fecha Finalización
Costo Real	Tiempo Real
Personal Asignado	

Formato: Solicitud de Paso a Producción

Nombre Empresa:		Formato para Solicitud de Paso a Producción	
Fecha de Elaboración: (AAA/MM/DD)		Fecha de Ejecución: (AAA/MM/DD)	
Elaborado por:		Ejecutado por:	
Proyecto/Tarea/Desarrollo:		Nro Solicitud:	
Objetivos del paso a producción:			
Paso de Programas:			
Servidor Origen	Servidor Destino	Programa	Esquema/Directorio
Tablas a crear o actualizar:			
Esquema	Tabla creada/modificada		
Procedimientos almacenados			
Esquema	Procedimiento almacenado		
Disparadores (triggers)			
Esquema	Nombre	Momento	Acción
Pasos para aplicar los cambios en producción:			
Paso	Ejecutado por	Hora	

Nota de aceptación

Presidente del Jurado

Jurado

Jurado

Ciudad y Fecha
